Edge Preserving Image Compression Technique Using Feed Forward Neural Network

A Thesis

Submitted in the partial fulfillment of requirements
for the award of the degree of

Master of Engineering

In

Electronic Instrumentation and Control Engineering

Submitted by
Sachchidanand Bajpayee
Regn.No 80751019

Under the esteemed guidance of

Mrs. GAGANDEEP KAUR
Lecturer(SG),EIED

Department of Electrical and Instrumentation Engineering
THAPAR UNIVERSITY
PATIALA (PUNJAB) 147004

July – 2009

CERTIFICATE

This is to certify that the thesis titled, "**Edge Preserving Image Compression Technique Using Feedforwad Neural Network**", being submitted by Sachchidanand Bajpayee, Regn. No. 80751019, in partial fulfillment for the requirements of the award of the degree of Master of Engineering in Thapar University, Patiala, is a record of student's own work carried out under my supervision and guidance and this thesis has not been submitted to any other university or institute for award of any degree.

Date 13/7/09

Sachchidanand bajpayee
Regn No. 80751019

It is certified that the above statement made by the student is correct to the best of our knowledge and belief.

Mentor & Supervisor
Mrs. Gagandeep kaur
Lecturer (S.G), EIED
Thapar University
Patiala

Dr. Smarajit Ghosh
Professor and Head
EIED, Thapar University
Patiala

ConutersignedBy 21/7
Dr. R.K.Sharma
Dean of Academic Affairs
Thapar University
Patiala

i

This thesis is dedicated to my parents

ACKNOWLEDGEMENT

Words are often too less to reveals one's deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis.

First of all I would like to thank the **Supreme Power,** one who has always guided me to work on the right path of the life. Without Her grace this would never come to be today's reality.

This work would not have been possible without the encouragement and able guidance of my supervisor, **Mrs Gagandeep kaur** their enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Their feedback and editorial comments were also invaluable for the writing of this thesis.

No words of thanks are enough for my **dear parents** whose support and care makes me stay on earth. Thanks to be with me.

At the end, I would like to thank all the faculty members of the department and my friends who directly or indirectly helped me in completion of my thesis.

Sachchidanand Bajpayee

(Regn No. 80751019)

ABSTRACT

With the growth of multimedia and internet, compression techniques have become the thrust area in the fields of computers. Popularity of multimedia has led to the integration of various types of computer data. Multimedia combines many data types like text, graphics, still images, animation, audio and video. Image compression is a process of efficiently coding digital image to reduce the number of bits required in representing image. Its purpose is to reduce the storage space and transmission cost while maintaining good quality. Many different image compression techniques currently exist for the compression of different types of images. In the present research work back propagation neural network training algorithm has been used. The neural network model has been trained and tested for the different types of images. Back propagation neural network algorithm helps to increase the performance of the system and to decrease the convergence time for the training of the neural network. The aim of this work is to develop an edge preserving image compressing technique using one hidden layer feed forward neural network of which the neurons are determined adaptively .The processed image block is fed as a single input pattern while single output pattern has been constructed from the original image unlike other neural network based technique where multiple image blocks are fed to train the network. The initialization of weights between the lone hidden layer by transforming pixel coordinates of the input pattern block into its equivalent one dimensional representation. The initialization process exhibit better rate convergence of the back propagation training algorithm as compare to the randomization of initial weight. The proposed scheme has been demonstrated through several experiments including lena, girl, cameraman and very promising results in compression as well as in reconstructed image over convectional neural network based technique are obtained.

CONTENTS

LIST OF FIGURE

LIST OF TABLE

LIST OF ABBREVIATIONS

CR	Compression ratio
PSNR	Peak signal to noise ratio
RMSE	Root mean square error
B/P	Bits per pixel
NN	Neural network
ANN	Artificial neural network
JPEG	Joint photographic expert group
MPEG	Moving picture expert group
GIF	Graphics interchange format
PNG	Portable network graphics
ECPS	Edge and corner preserving smoother
LMS	Least mean square
ART	Adaptive resonance theory
MCP	McCulloch and pitts model
EA	Error derivative

Literature survey

Abbas Rizwi(1992) introduced an image compression algorithm with a new bit rate control capability. The bit rate control technique is developed for use in conjunction with the PEG baseline image compression algorithm. This new method was an extension of the previously developed algorithm which is implemented in the Zoran image compression chip set. The chip set is comprised of a Discrete Cosine Transform Processor and an Image Compression Coder/Decoder. Simulations were performed on several images using the Zoran bit rate control incorporated in the JPEG algorithm and the iterative algorithm described in the ISO/JTCl/SC2/WG8 N 800 report. The iterative algorithm uses the Newton Raphson method to converge to an optimal scale factor to obtain the desired bit rate. The Zoran bit rate control algorithm utilized in conjunction with the baseline JPEG algorithm uses two pass compression. The main disadvantage of the iterative process is the number of Passl's performed to obtain the optimal scale factor for a given target Bit Per Pixel, prior to performing the actual encoding pass. The Zoran bit rate control technique incorporated in the JPEG algorithms, performs only one preliminary pass prior to Pass2 [1].

Ronald .A.Devore(1992) proposed a new theory for analyzing image compression methods that are based on compression of wavelet decompositions. This theory precisely relates the rate of decay in the error between the original image and the compressed image (measured in one of a family of so called Lp norms) as the size of the compressed image representation increases (i.e. as the amount of compression decreases) to b) the smoothness of the image in certain smoothness classes called Besov spaces, within this theory, the error incurred by the quantization of wavelet transform coefficients is explained. Several compression algorithms based on piecewise constant approximations are analyzed. It is shown that if pictures can be characterized by their membership in the smoothness classes considered here, then wavelet based methods are near optimal within a larger class of stable (in a particular mathematical sense) transform based, nonlinear methods of image compression [2].

David Jeff Jackson et.al (1993) addressed the area of data compression as it is applicable to image processing. An analysis of several image compression strategies are examined for their relative effectiveness. Several topics concerning image compression are examined in this study including generic data compression algorithms, file format schemes and fractal image compression. An overview of the popular LZW compression algorithm and its subsequent variations is also given. A survey of several common image file formats is presented with respect to the differing approaches to image compression. Fractal compression is examined in

depth to reveal how an interactive approach to image compression is implemented. A comparison of the previously mentioned techniques was performed using several sources. The original GIF images were used to construct the JPEG formatted files [3].

P. Moravie et.al(1995) emphasized that in the digitized satellite image domain, the needs for high dimension images increase considerably. To transmit or to stock such images (more than 6000 by 6000 pixels), we need to reduce their data volume and so we have to use image compression technique. In most cases, these operations have to be processed in Real Time. The large amount of computations required by classical image compression algorithms prohibits the use of common sequential processors. To solve that problem, CEA in collaboration with CNES has tried to define the best suited architecture for the image compression. In order to achieve that aim, he developed and evaluated a new parallel image compression algorithm for general purpose parallel computers using data parallelism. A new parallel image compression algorithm which is able to be implemented on SIMD and MIMD architecture [4].

J Jiang (1995) proposed an extensive survey on the development of neural network technology for image compression. Three main categories had been covered .These includes neural networks directly developed for image compression, neural network implementation of traditional algorithms and development of neural network based technology which provide further improvements over the existing image compression algorithms. He developed various up to date neural network technologies in image compression and coding. These include direct development of neural learning algorithms and indirect applications of neural networks to improve existing image compression techniques. For direct learning algorithm, development of neural vector quantization stands out to be the most promising technique which has shown improvement over traditional algorithms. As vector quantization was included in many image compression algorithms such as JPEG, MPEG, and wavelets based variables etc, many practical applications have been initiated in commercial world [5].

Wayne *0.* Cochran et.al(1996) proposed a extension of fractal image compression to volumetric data was trivial in theory. The simple addition of a dimension to existing fractal image compression algorithms results in infeasible compression times and noncompetitive volume compression results. This paper extends several fractal image compression enhancements to perform properly and efficiently on volumetric data, and introduces a new 3D edge classification scheme based on principal component analysis. Numerous experiments over the many parameters of fractal volume compression suggest aggressive settings of its system parameters. At that peak efficiency, fractal volume compression

surpasses vector quantization and approaches within 1 dB PSNR of the discrete cosine transform. When compared to the DCT, fractal volume compression represents surfaces in volumes exceptionally well at high compression rates, and the artifacts of its compression error appears as noise instead of deceptive smoothing or distracting ringing. Fractal image compression extends simply and directly to three dimensions, but will not perform adequately without special volumetric enhancements. Principal component analysis classification of 3D volume blocks, and a down sampled nearest neighbor search combine to reduce volume compression time from hours to minutes, with minor impact on compression fidelity. Further reduction of the domain pool to macro blocks further increases the compression rate with a negligible impact on fidelity, and allows fractal coded volumes to be rendered directly from the compressed representation. The additional dimension of fractal volume compression produces a richer domain pool, resulting in higher compression rates than its 2D image counterpart [6].

Maire D. Reavy et.al(1997) illustrated a new method of lossless bi level image compression introduced to replace JBIG and G3, the current standards for bi level and facsimile image compression. This paper applies the BACIC (Block Arithmetic Coding for Image Compression) algorithm to reduced grayscale and full grayscale image compression. BACIC's compressed files are slightly smaller than JBIG's, twice as small as G3's, and at least thirty percent smaller than Lossless JPEG's (when Lossless JPEG uses Huffman coding) for reduced grayscale images with fewer than 7 bits/pixel. In it he introduced BACIC, a new method for lossless compression of bi level images, as an efficient lossless compression of reduced grayscale images [7].

S. W.Hong et.al(2000) proposed an edge preserving image compression model is presented, based on sub band coding and iterative constrained least square regularization. The idea is to incorporate the technique of image restoration into the current lossy image compression schemes. The model utilizes the edge information extracted from the source image as *a* priori knowledge for the subsequent reconstruction. Generally, the extracted edge information has *a* limited range of magnitudes and it can be lossily conveyed. Subband coding, one of the outstanding lossy image compression schemes, is incorporated to compress the source image. Vector quantization, a block based lossy compression technique, is employed to compromise the bit rate incurred by the additional edge information and the target bit rate. Experiments had shown that the approach could significantly improve both the objective and subjective quality of the reconstructed image by preserving more edge details. Specifically, the model incorporated with SPIHT (set partitioning in hierarchical trees) outperformed the original

SPIHT with the 'Baboon' continuous tone test image. In general, the model may be applied to any lossy image compression systems [8].

Mitchell A. Golner et.al(2000) discussed concepts that optimized image compression ratio by utilizing the information about a signal's properties and their uses. This additional information about the image is used to achieve further gains in image compression. The techniques developed in this work are on the ubiquitous JPEG Still Image Compression Standard [IS0941 for compression of continuous tone grayscale and color images. This paper is based on a region based variable quantization JPEG software codec that was developed tested and compared with other image compression techniques. The application, named JPEG Tool, has a Graphical User Interface (GUI) and runs under Microsoft Windows 95. This paper discussed briefly the standard JPEG implementation and lays foundation for software extension to the standard. Region selection techniques and algorithms that complement variable quantization techniques are presented in addition to a brief discussion on the theory and implementation of variable quantization schemes. The paper includes a presentation of generalized criteria for image compression performance and specific results obtained with JPEG Tool**.** The research on a variable quantization technique has led to the development of linear and raised cosine interpolation. The use of a wide range of scaling factors has resulted in a greater difference in fidelity between blocks coded with highest and lowest fidelities. This is enabled by the use of interpolation in the variable quantization mask. The region selection can be performed manually or automatically according to predetermined requirements [9].

Clark N. Taylor et.al(2001) proposed the ubiquitous wireless multimedia communication, the bottlenecks to communicating multimedia data over wireless channels must be addressed. Two significant bottlenecks which need to be overcome are the bandwidth and energy consumption requirements for mobile multimedia communication. In this paper, he addressed the bandwidth and energy dissipation bottlenecks by adapting image compression parameters to current communication conditions and constraints. We focus on the JPEG image compression algorithm, and present the results of varying some image compression parameters on energy dissipation bandwidth required, and quality of image received. He presents a methodology for selecting the JPEG image compression parameters in order to minimize energy consumption while meeting latency, bandwidth, and quality of image constraints**.** By adapting the source coder of a multimedia capable radio to current communication conditions and constraints, it is possible to overcome the bandwidth and energy bottlenecks to wireless multimedia communication. He have selected two parameters

of the JPEG image compression algorithm to vary, and presented the results of modifying the parameters on quality of image, bandwidth required, computation energy, and communication energy. He has also presented a methodology which run time selects the optimal JPEG parameters to minimize overall energy consumption, helping to enable wireless multimedia communication [10].

Pou Yah Wu(2001) emphasized on the distributed fractal image compression and decompression on the PVM system. He applied the regional search for the fractal image compression to reduce the communication cost on the distributed system PVM. The regional search is to search the partitioned iterated function system from a region of the image instead of over the whole image. Because the area surrounding of a partitioned block is similar to this block possibly, finding the fractal codes by regional search has a higher compression ratio and less compression time. When implemented on the PVM**,** the fractal image compression using regional search reduces the compression time with less compression loss. When he compress the image Lena with image size 1024 x 1024 using the region size 512 x 512 on the PVM with 4Pentiuni 11 300 PCs, the compression time is 13.6 seconds, the compression ratio is 6.34 and the PSNR is 38.59. But it needs to take 176 seconds, have the compression ratio 6.30 and have PSNR 39.68 by the conventional fractal image compression. In addition, when the region size is I28 x I28, the compression time is 7.8 seconds, the compression ratio is 7.53 and the PSNR is 36.67. In the future, we can apply this method to the fractal image compression using neural networks [11].

Koon Pong Wong(2002) in his paper discussed the increasing use of teleradiology systems, large amount of data is acquired and transmitted, thus raising the issue of medical image compression. The goal of image compression is to represent an image with as few number of bits as possible while preserving the quality required for the given application. Standard lossless compression schemes can only yield compression ratios of about 2 1 that are insufficient to compress volumetric tomographic image data. Common standards available in industry for lossy image compression are usually used in non medical applications and their application to medical image compression is still under investigation. Fractal image coding is a new compression technique that has received much attention recently. This study investigates the feasibility of applying fractal image coding approach to nuclear medicine image compression. Quadtree based partition scheme was used in the encoding phase to partition an image frame into small irregular segments that after decoding process yield an approximate image to the original. His preliminary results showed that compression ratios higher than 10:1 can be achieved in clinical images. It was also found that there is no

diagnostic loss in the parametric images computed from the reconstructed images as compared to those obtained from the original raw data. He concluded that fractal image coding could be used to compress tomographic images and it may be useful in telemedicine [12].

Michael T.Kurdziel (2002) proposed that HF communication channel was notorious for its degraded channel including low signal to noise ratio, Doppler and multi path spreading and high level of interference. Image transmission over HF radio system could particularly challenging the size of some digital image. This paper presents some novel image compression technique coupled with three HF transmission schemes. The image compression scheme is based on a phase dispersion technique originally developed for the design of pulse compression radar waveforms. This technique forms the basis for a lossless, invertible transform that converts sub band coefficient to an intermediate domain. This image compression scheme generates significance map bit which must be transferred across the HF link error free as well as magnitude vectors, which can sustain some degree of received error [13].

Aaron T. Deever et.al(2003) laid the emphasis on Reversible integer wavelet transforms are increasingly popular in lossless image compression, as evidenced by their use in the recently developed JPEG2000 image coding standard. In his paper, a projection based technique is presented for decreasing the first order entropy of transform coefficients and improving the lossless compression performance of reversible integer wavelet transforms. The projection technique is developed and used to predict a wavelet transform coefficient as a linear combination of other wavelet transform coefficients. It yields optimal fixed prediction steps for lifting based wavelet transforms and unifies much wavelet based lossless image compression. Additionally, the projection technique is used in an adaptive prediction scheme that varies the final prediction step of the lifting based transform based on a modeling context. Compared to current fixed and adaptive lifting based transforms, the projection technique produces improved reversible integer wavelet transforms with superior lossless compression performance. It also provides a generalized framework that explains and unifies many previous results in wavelet based lossless image compression [14].

Deepti Gupta et.al(2003) showed that wavelet transform and quantization methods have produced the algorithm capable of surpassing the existing image compression standards like the Joint Photographic Experts Group (JPEG Algorithm). His paper presents new wavelet packets algorithms for image compressions. Extensive experiment results prove that our techniques exhibit performance equal to, or in several cases superior to, the current wavelet

filters. Wavelet transform is modeling the signals by combining algorithm based on wavelet. The Advances in translations and dilations of a simple oscillatory function of finite duration called a wavelet. Wavelet transform is used for analysis of the image at different decomposition levels [15].

G. Y. Chen et.al(2004) shown that Wavelets have been successfully used in image compression .However, for the given image, the choice of the wavelet to use is an important issue. In his paper he proposes to use optimal wavelet for image compression, given the number of most significant wavelet coefficients to be kept. Simulated Annealing is used to find the optimal wavelet for the given image to be compressed. In Simulated Annealing, he needed a cost function to minimize. This cost function is defined as the mean square error between the decompressed image and the original image. He conduct some experiments in Matlab by using the test images Lena MRIS can and Fingerprint. These images are available in WaueLab developed by Donoho et al. at Stanford University. Experimental results show that this approach is better than the Daubechies 8 wavelet (08) or image compression. In some cases, we get nearly 0.6dB improvement over D8 by using optimal wavelet. This indicates that the choice of the wavelet indeed makes a significant difference in image compression [16].

Mei Tian et.al(2005) discusses the possibility of Singular Value Decomposition in Image Compression applications. A mistake viewpoint that is about SVD based image compression scheme is demonstrated. His paper goes deep to study three schemes of SVD based image compression and prove the usage feasibility of SVD based image compression**.** In his paper we investigate into using singular value decomposition as a method of image compression scheme. The application of SVD based image compression schemes has shown their effectiveness. Although he has drawn some progress, some limitations of his study are still in existence. These limitations could be solved by our future work. The percentage of the sum of the singular values should be flexible selected according to different images and adaptive to different sub block of the same image [17].

Erjun Zhao(2005) described that Fractal image compression is a new technique in image compression field based on Affine contractive transforms. Fractal image compression methods belong to different categories according to the different theories they are based on. All of those are discussed in this paper. In the end a conclusion is made to summarize the characters of fractal image compression [18].

Sheng Liu et.al(2006) had worked in the field of radar image compression of voyage data recorder (VDR). A sheet of radar image is storage in VDR after an interval of time, so the

compression algorithm for radar image may be seen as immobile image compression. The image compression process includes Discrete Wavelet Transform (DWT), quantization and entropy coding. First, the DWT and its fast Mallat algorithm is presented. Then, the character of the image after DWT is analyzed. The Set partitioning in hierarchical trees (SPIHT) coder includes the function of quantization and entropy coding. The SPIHT coder is explained in his paper briefly. At last several wavelet functions in common use are chosen to compress the radar image and code for it with two coding algorithms embedded zero tree wavelet (EZW) and SPIHT. The simulation results show the SPIHT is more effective than EZW [19].

Xiangjian He(2006) laid the emphasis on distributed and network based pattern recognition. For real time object recognition or reconstruction, image compression can greatly reduce the image size, and hence increase the processing speed and enhance performance. Fractal image compression is a relatively recent image compression method. Its basic idea was to represent images as a fixed point of a contractive Iterated Function System (IFS). Spiral Architecture (SA) is a novel image structure on which images are displayed as a collection of hexagonal pixels. The efficiency and accuracy of image processing on SA had been demonstrated in many published papers. He had shown the existence of contractive IFS's through the construction of a Complete Metric Space on SA. The selection of range and domain blocks for fractal image compression is highly related to the uniform image separation specific to SA [20].

Ivan Vilovic(2006) proposed that there are number of trials of using neural networks as signal processing tools for image compression. A direct solution method is used for image compression using the neural networks. An experience of using multilayer perceptron for image compression is presented. The multilayer perceptron is used for transform coding of the image. The network is trained for different number of hidden neurons with direct impact to compress ratio is experimented with different images that have been segmented in the blocks of various sizes for compression process. Reconstructed image is compared with original image using signal to noise ratio and number of bits per pixel [21].

Tao Wang et.al(2006) introduced a light path structure and principal of optical wavelet transform. It is realized by using 4f optical information processing system. As optical wavelet transform utilizes the parallel computation of optical elements, it features high conversion speed. And it is applied extensively in vision systems such as pattern recognition, image feature extraction, image edge enhancement etc. But there is no obvious progress in the research of image compression. In his paper, he analyzes the requirements of optical system in image compression based on optical wavelet transform. The requirements are mainly about

the performance of light source, light path structure, optical elements and filter. And some improved measurements are put forward according to present problems. First, the method of determining focuses of lens and how to use Talbot effect to locate the object plane and spectrum plane are given. Second, the use of reflecting 4f system is put forward to decrease chromatic aberration and noise in optical wavelet transform. Third, adopt liquid crystal light valve and spatial light modulator to strengthen flexibility and practicability. Finally, we analyze the merits of using white light information processing system for image compression [22**].**

Kin Wah Ching Eugene et.al(2006) proposed an improvement scheme, so named the Two Pass Improved Encoding Scheme (TIES), for the application to image compression through the extension of the existing concept of Fractal Image Compression (FIC), which capitalizes on the self similarity within a given image to be compressed. We first briefly explore the existing image compression technology based on FIC, before proceeding to establish the concept behind the TIES algorithm. We then devise an effective encoding and decoding algorithm for the implementation of TIES through the consideration of the domain pool of an image, domain block transformation scaling and intensity variation, range block approximation using linear combinations, and finally the use of an arithmetic compression algorithm to store the final data as close to source entropy as possible [23].

Yukinari Nishikawa et.al(2007) described a high speed CMOS IMAGE sensor with on chip parallel image compression circuits. This chip consists of a pixel array, an A/D converter array with noise canceling function and an image compression processing element array and buffer memories. The image compression processing element is implemented with a 4_4 point Discrete Cosine Transform (DCT) and a modified zigzag scanner with 4 blocks. A prototype high speed CMOS image sensor integrating the image compression circuits is implemented based on 1 poly 5 metal 0.25m CMOS technology. Image encoding using the implemented parallel image compression circuits to the image captured by the high speed image sensor is successfully performed at 3,000[frame/s] [24].

Jian Li et.al(2007) proposed a quadtree partitioning fractal image compression (QPFIC) method used for the partial discharge (PD) image remote recognition system. Self similarity in PD images is the premise of fractal image compression and is described for the typical PD images acquired from defect model experiments in laboratory. Influences of fractal image compression on a group of PD image features are discussed. Fifty PD data samples are used to qualify the QPFIC to be used in remote PD pattern recognition. Analysis results show that

the QPFIC method produces errors of the computational features. Such errors could not influence the PD image recognition results under the control of the PD image compression errors. His paper describes the similarity existing in PD images presents the influences by the quad tree partitioning fractional image compression method on the recognition of decoded PD images [25].

Li Ke et.al(2007) described that when wavelet transform is applied to image compression, the chosen wavelet base affects the efficiency of coding and the quality of the reconstructed image, because the property parameters of different wavelet bases are varied, it is very important to research the correlation between the wavelet base properties and image compression. Chosen wavelet bases of one family for the experiments are convenient for analyzing contrastively, and the results have high reliability, and Daubechies wavelet bases with the properties of compactly supported, orthogonality, regularity, vanishing moment are widely used, then the paper chooses Daubechies wavelet bases as the research object, analyzes the correlation between the wavelet base properties and the image compression. The experiment results present the principles of wavelet base choice in image compression [26].

Jian Li et.al(2008) introduced a quadtree partitioning fractal image compression method used for the partial discharge (PD) image remote recognition system. Self similarity in PD images is the premise of fractal image compression and is described for the typical PD images acquired from defect model experiments in laboratory. Influences of fractal image compression on a group of PD image features are discussed. Fifty PD data samples are used to qualify the QPFIC to be used in remote PD pattern recognition. Analysis results show that the QPFIC method produces errors of the computational features. Such errors could not influence the PD image recognition results under the control of the PD image compression errors.

CHAPTER 1

INTRODUCTION

Image compression is a process of efficiently coding digital image, to reduce the number of bits required in representing image. Its purpose is to reduce the storage space and transmission cost while maintaining good quality. A number of neural network based image compression scheme have been proposed for this purpose, which are classified as lossless or lossy image compression technique. Rutherhart in his early work explored the potential of neural network to achieve data encoding /decoding, which later utilized by many researches for image compression employing the standard back propagation training algorithm. In most of the methods , an image is divided into number of non overlapping pixel blocks, and fed as patterns for network training .Image compression is achieved by encoding the pixel blocks into the trained weight set ,which is transmitted to the receiving side for reconstruction of the image

In comparison with the vector quantization, this method has certain advantage because here no utilization of code books are required and encoding/decoding time are much less. But in such cases very limited amount of compression is achieved since it exploited only the correlation between pixel within each of the training patterns .Higher compression ratio was achieved in by developing hierarchical NN that cost heavily due to the physical structure of the NN. Adaptive one hidden layer feed forward neural network has been developed for successful image compression that reduces the network size as well as the computational time based on the image size. But none of the above method considers the difficulty of tackling the huge image size during training .Generally depending upon the nature of the image to be compressed there are two basic possibilities of this approach , first shorter image blocks results huge number of training pattern and second bigger image block results huge dimensions of the training pattern. Therefore each of these cases is difficult to handle with respect to training time as well as physical structure of the network.

To make image compression practical, it is mandatory to reduce the huge size of most image data that eventually reduces physical structure of the NN. In order to reduce the size considerable several image processing steps namely edge detection, thresholding, thinning are applied on the

image and discussed briefly. The main concern of the second phase of the work is to adaptively determine the structure of the NN that encodes the image using back propagation training method . The basic aim is to develop an edge preserving image compression technique using one hidden layer feed forward neural network of which the neurons are determined adaptively based on the images to be compressed .Edge detection is important data reduction step since it encodes information based on the structure of the image. Using edge detection vital information of the image is preserved while keeping aside less important information that effectively reduces dynamic range of the image and elements pixel redundancy. As a next step the image is thresholded to detect the pixel having less influence on the image and therefore removed. A threshold function has been designed using gray level information of the edge detected image and applied to reduce the size further. Finally thining operation has been applied based on the interpolation method to reduce thickness of the image. Now vital information has been preserved in the single image block (PIB) while its size has been reduced significantly and fed as a single input pattern the NN. It is worth to mention here that processing never destroy spatial information of the original image which has been stored along with the pixel values. The number of pixels present in the PIB determines number of input and output neurons of the NN .Input pattern vector has been framed using gray level information of the pixels of the PIB while output pattern vector is constructed using gray level information of the pixel taking from the original image having same spatial coordinates of the PIB. The training pattern thus generated image dependent and is flexible enough compare to the previous NN based technique where output pattern are just replica of the input patterns. The input and output pattern vectors are normalized within and are fed to the input and output layer of the NN. Neurons at the hidden layer are determined adaptively and set to the number of distinct gray level present in the PIB, which is surely less than the number of input/output neurons. The inputs neurons representing the same gray values are connected with the output neurons representing the same gray value that of input. Thus, adaptively the network has been formed and provides modular structure, which facilitates fault detection and less susceptible to failure.

A new technique has been adopted in the paper while initializing the weight between input and hidden layer neurons instead of randomizing the initial weight ,here spatial coordinates of the pixel of the image block are converted from two to one dimensional value and normalized with in [0,1].This approach demonstrate fast rate of convergence of the training algorithm and has

been tested for a number of images including lena. The network has been trained using standard back propagation algorithm for a fixed time period determined a priori and at the end of training the computed output of the hidden layer neurons represent the compressed image. The trained weight, computed output of the hidden neurons threshold and the coordinates of the PIB are transmitted to the receiving side for reconstructing image, which are together much less than the original image size. At the receiving side, a network is formed for decompression of the compressed image. Weights between these two layers are set to the trained weight and after a single forward pass the image is reconstructed, very similar to the original image. The transport of images across communication paths is an expensive process. Image compression provides an option for reducing the number of bits in transmission. This in turn helps increase the volume of data transferred in a space of time, along with reducing the cost required. It has become increasingly important to most computer networks, as the volume of data traffic has begun to exceed their capacity for transmission. Traditional techniques that have already been identified for data compression includes Predictive coding, Transform coding and Vector Quantization. In brief, predictive coding refers to the de correlation of similar neighboring pixels within an image to remove redundancy. Following the removal of redundant data, a more compressed image or signal may be transmitted. Transform based compression techniques have also been commonly employed.

These techniques execute transformations on images to produce a set of coefficients. A subset of coefficients is chosen that allows good data representation (minimum distortion) while maintaining an adequate amount of compression for transmission. The results achieved with a transform based technique is highly dependent on the choice of transformation used (cosine, wavelet, Karhunen Loeve etc). Finally, vector quantization techniques require the development of an appropriate codebook to compress data. Usage of code books do not guarantee convergence and hence do not necessarily deliver infallible decoding accuracy. Also the process may be very slow for large codebooks as the process requires extensive searches through the entire codebook. Following the review of some of the traditional techniques for image compression, it is possible to discuss some of the more recent techniques that may be employed for data compression.

Artificial Neural Networks (ANNs) have been applied to many problems, and have demonstrated their superiority over traditional methods when dealing with noisy or incomplete data. One such

application is for image compression. Neural networks seem to be well suited to this particular function, as they have the ability to pre process input patterns to produce simpler patterns with fewer components. This compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. Not only can ANN based techniques provide sufficient compression rates of the data in question, but security is easily maintained. This occurs because the compressed data that is sent along a communication line is encoded and does not resemble its original form. There have already been an exhaustive number of papers published applying ANNs to image compression. Many different training algorithms and architectures have been used. Some of the more notable in the literature are nested training algorithms used with symmetrical multilayer neural networks, Self organizing maps, for codebook generation, principal component analysis networks, back propagation networks, and the adaptive principal component extraction algorithm. Apart from the existing technology on image compression represented by series of JPEG, MPEG and H.26x standards, new technology such as neural networks and genetic algorithms are being developed to explore the future of image coding. Successful applications of neural networks to vector quantization have now become well established, and other aspects of neural network involvement in this area are stepping up to play significant roles in assisting with those traditional technologies.

1.1 Types of Image Compression

JPEG stands for Joint Photographic Experts Group, the original name of the Committee that wrote the standard. JPEG is designed for compressing full color or Gray scale images of natural, real world scenes. It works well on photographs, naturalistic artwork, and similar material; not so well on lettering, simple cartoons, or line drawings. JPEG handles only still images, but there is a related standard called MPEG for motion pictures. JPEG is "lossy," meaning that the decompressed image isn't quite the same as the one you started with. (There are lossless image compression algorithms, but JPEG achieves much greater compression than is possible with lossless methods.) JPEG is designed to exploit known limitations of the human eye, notably the fact that small colour changes are perceived less accurately than small changes in brightness. Thus, JPEG is intended for compressing images that will be looked at by humans. If you plan to machine analyze your images, the small errors introduced by JPEG may be a problem for you, even if they are invisible to the eye.

The Graphics Interchange Format (GIF) is an 8 bit per pixel bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability. The format uses a palette of up to 256 distinct colors from the 24 bit RGB color space. It also supports animations and allows a separate palette of 256 colors for each frame. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with continuous color, but it is well suited for more simple images such as graphics or logos with solid areas of color. GIF images are compressed using the Lempel Ziv Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality. This compression technique was patented in 1985. Controversy over the licensing agreement between the patent holder, Unisys, and CompuServe in 1994 inspired the development of the Portable Network Graphics (PNG) standard; since then all the relevant patents have expired.

Fractal compression is a lossy image compression method using fractals to achieve high levels of compression. The method is best suited for photographs of natural scenes (trees, mountains, ferns, clouds). The fractal compression technique relies on the fact that in certain images, parts of the image resemble other parts of the same image. Fractal algorithms convert these parts, or more precisely, geometric shapes into mathematical data called "fractal codes" which are used to recreate the encoded image. Fractal compression differs from pixel based compression schemes such as JPEG, GIF and MPEG since no pixels are saved. Once an image has been converted into fractal code its relationship to a specific resolution has been lost; it becomes resolution independent. The image can be recreated to fill any screen size without the introduction of image artifacts or loss of sharpness that occurs in pixel based compression schemes.

1.2 Learning in Neural Networks

Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place. All learning methods used for neural networks can be classified into two major categories

Supervised Learning which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global

information may be required. Paradigms of supervised learning include error correction learning (back propagation algorithm), reinforcement learning and stochastic learning.

Unsupervised Learning uses no external teacher and is based upon only local information. It is also referred to as self organization, in the sense that it self organizes data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and competitive learning.

1.3 Advantages of Neural Networks

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self Organization: An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage

1.4 Overview of Back Propagation Algorithm

Minsky and Papert (1969) showed that there are many simple problems such as the exclusive or problem which linear neural networks can not solve. Note that term "solve" means learn the desired associative links. Argument is that if such networks can not solve such simple problems how they could solve complex problems in vision, language, and motor control. Solutions to this problem were as follows:

- Select appropriate "recoding" scheme which transforms inputs.
- Perceptron Learning Rule Requires that you correctly "guess" an acceptable input to hidden unit mapping.
- Back propagation learning rule Learn both sets of weights simultaneously.

Back propagation is a form of supervised learning for multi layer nets, also known as the generalized delta rule. Error data at the output layer is "back propagated" to earlier ones, allowing incoming weights to these layers to be updated. It is most often used as training algorithm in current neural network applications. The back propagation algorithm was developed by Paul Werbos in 1974 and rediscovered independently by Rumelhart and Parker. Since its rediscovery, the back propagation algorithm has been widely used as a learning algorithm in feed forward multilayer neural networks. What makes this algorithm different than the others is the process by which the weights are calculated during the learning network. In general, the difficulty with multilayer Perceptrons is calculating the weights of the hidden layers in an efficient way that result in the least (or zero) output error; the more hidden layers there are, the more difficult it becomes. To update the weights, one must calculate an error. At the output layer this error is easily measured; this is the difference between the actual and desired (target) outputs. At the hidden layers, however, there is no direct observation of the error; hence, some other technique must be used. To calculate an error at the hidden layers that will cause minimization of the output error, as this is the ultimate goal. The back propagation algorithm is an involved mathematical tool; however, execution of the training equations is based on iterative processes, and thus is easily implementable on computer.

1.5 Objective of Thesis

The objectives of thesis are

- Exploration of a supervised learning algorithm for artificial neural networks i.e. the Error Back propagation learning algorithm for a layered feed forward network.
- Implementation of the Back Propagation learning algorithm in image compression.
- Analysis of the simulation results of Back Propagation algorithm.

Chapter 2

Image Compression

2.1 Image Compression

Image compression means minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more image to be stored in a given amount of disk more memory space. It also reduces the time required for image to be sent over the internet or downloaded from web pages. Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and bandwidth. Despite rapid progress in mass storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data transmission bandwidth continues to outstrip the capabilities of available technology. The recent growth of data intensive multimedia based web application have not only sustained the need for more efficient ways to encode signals and images but have made compression of such signal central to storage and communication technology.

2.2 Need for Compression

An image 1024 pixel *1024 pixel *1024 pixel *24 bit , without compression would require 3 MB of storage and 7 minute of transmission, utilizing a high speed , 64 kbs, ISDN line . If the image is compressed at a 10:1 compression ratio, the storage requirement is reduced to 300 KB and the transmission time is reduced to less than 7 second. Images file in an uncompressed form are very large, and the internet especially for people using a 56 kbps dialup modem, can be pretty slow. This combination could seriously limit one of the webs most appreciated aspects –its ability to present images easily.

The figure in table show qualitative transition from simple text to full motion video data and the disk space, transmission bandwidth and the transmission time needed to store and transmit such uncompressed data .

Table 2.1 Multimedia data types and uncompressed storage space, transmission bandwidth, and transmission time required

Multimedia data	Size/ duration	Bits/pixel Or bits/sample	Uncompressed Size (B for bytes)	Transmission Bandwidth (B for bits)	Transmission time (using a 28.8k modem)
A page of text	11'' ×8.5"	Varying solution	4 8 KB	3264kb/pages	1.1 2.2 sec
Telephone quality speech	10 sec	8 bps	80 KB	64 kb /sec	22.2 sec
Gray scale image	512×512	8 bpp	262 KB	2.1 MB/image	1 min 13 sec
Color image	512 ×512	24 bpp	786 KB	6.29Mb/image	3 min 39 sec
Medical image	2048×1680	12 bpp	5.16 Mb	41.3Mb/image	23 min 54 sec
SHD image	2048× 2048	24 bpp	12.58 MB	100MB/image	58 min 15 sec
Full motion Video	640×480,1min	24 bpp	1.66 GB	211 Mb/sec	5days 8 hours

The examples above clearly illustrate the need for sufficient storage space , large transmission bandwidth, and long transmission time for image . Audio and video data at present, the only solution is to compress multimedia data before its storage and transmission and decompress it at the receiver for play back for example with a compression ratio of 32 1, the space , bandwidth and the transmission time requirement can be reduced by a factor of 32 , with acceptable quality.

2.3 Types of Compression

In the case of video, compression ratio causes some information to be lost ; some information at a detailed level is considered not essential for a reasonable reproduction of scene. This type of compression is called lossy compression, audio compression on the other hand is not lossy, it is called loss less compression.

2.3.1 Lossless Compression

In lossless compression scheme, the reconstructed image after compression is numerically identical to original image. However lossless compression can only achieve a modest amount of compression. Lossless coding guaranties that the decompressed image is absolutely identical to the image before compression. This is an important requirement for some application domain, e.g. medical imaging, where not only high quality is in demand but unaltered archiving is a legal requirement. Lossless technique can also be used for the compression of other data types where loss of information is not acceptable, e.g. text document and program executables.

2.3.2 Lossy Compression

Lossy is a term applied to data compression technique in which some amount of the original data is lost during the compression process. Lossy image compression applications attempt to eliminate redundant or un necessary information in terms of what the human eye can perceive. An image reconstructed following lossy compression contains degradation relative to the original image. Often this is because the compression scheme completely discards redundant information. however lossy scheme are capable of achieving much higher compression. Under normal viewing condition, no visible loss is perceived (visually lossless).

Lossy image data compression is useful for application to the world wide images for quicker transfer across the internet. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discard the redundant information.

2.4 Applications of Compression

Applications of data compression are primarily in transmission and in storage of information. Image transmission application are in broadcast television remote sensing via satellite, military application via aircraft, radar and sonar, teleconferencing, computer communication, facsimile transmission, etc. Image storage is required for educational business documents, medical images, that rises in computer tomography, magnetic resonance imaging and digital radiology, motion picture, satellite image weather maps, etc. Application of data compression is also possible in the development of fast algorithm, where the number of operations required to be implement an algorithm is reduced by working with compressed data.

Over the year, the need for image compression has grown steadily. Currently it is recognized as an Enabling technology. It plays a crucial role in many important and diverse applications such as:

1. Business document, where lossy compression is prohibited for legal reasons.
2. Satellite image where the data loss is undesirable because of image collect cost.
3. Medical image where difference in original image and uncompressed one can compress diagnostic accuracy.
4. Tele -video conferencing.
5. Remote sensing.
6. Space and hazardous waste water application .
7. Control of remotely piloted vehicle in military.
8. Facsimile transmission (fax)

2.5 Characteristic to Judge Compression Algorithm

Image quality describes the fidelity with which an image compression scheme recreates the source image data. There are four main characteristics to judge image compression algorithms

1. Compression Ratio
2. Compression Speed
3. Mean Square Error
4. Peak Signal to Noise Ratio

These characteristics are used to determine the suitability of a given compression algorithm for any application.

2.5.1 Compression Ratio

The compression ratio is equal to the size of the original image divided by the size of the compressed image. This ratio gives how much compression is achieved for a particular image. The compression ratio achieved usually indicates the picture quality. Generally, the higher the compression ratio, the poorer the quality of the resulting image. The trade off between compression ratio and picture quality is an important one to consider when compressing images. Some compression schemes produce compression ratios that are highly dependent on the image content. This aspect of compression is called data dependency. Using an algorithm with a high degree of data dependency, an image of a crowd at a football game (which contains a lot of

detail) may produce a very small compression ratio, whereas an image of a blue sky (which consists mostly of constant colors and intensities) may produce a very high compression ratio.

2.5.2 Compression Speed

Compression time and decompression time are defined as the amount of time required to compress and decompress a picture, respectively. Their value depends on the following considerations:

- The complexity of the compression algorithm.
- The efficiency of the software or hardware implementation of the algorithm.
- The speed of the utilized processor or auxiliary hardware.

Generally, the faster that both operations can be performed, the better. Fast compression time increases the speed with which material can be created. Fast decompression time increases the speed with which the user can display and interact with images.

2.5.3 Root Mean Square Error

Mean square error measures the cumulative square error between the original and the compressed image.

The formula for mean square is given as

$$\text{RMSE} = \left[\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\hat{f}(i,j) - f(i,j) \right]^2 \right]^{1/2} \tag{2.1}$$

Where M*N is the size of the image, $\hat{f}(i, j)$ and $f(i, j)$ are the matrix element of the decompressed and the original image at (i,j) pixel.

2.5.4 Peak Signal to Noise Ratio

Peak signal to reconstructed image measure known as PSNR.

$$\text{PSNR} = 10 \log_{10} \left[\frac{M \times N}{RMSE^2} \right]$$

Here signal is original image and noise is error in reconstructed image.

In general, a good reconstructed image is one with low MSE and high PSNR. That means that the image has low error.

2.6 Edge Preserving and Smoothing

Edge preserving smoothing is an image processing technique where the edge information is preserved during the smoothing process. It uses non linear operator which is able to remove texture and noise, while keeping edges and corners. The technique is known also as edge and corner preserving smoother (ECPS). Smoothing is an important task in image processing. The best known smoothing technique is low pass linear filtering. The most widely used filter deploys a Gaussian function as a smoothing kernel. However, since linear low pass filtering strongly attenuates high frequency components, not only noise, but also edges and corners, are smoothed. Therefore, there has been a remarkable effort to develop nonlinear operators able to remove texture and noise while preserving edges and corners. In the following we refer to such an operator as an Edge and Corner Preserving Smoother (ECPS). Several ECPSs have been proposed in the literature. The best known ones are based on median filtering, morphological analysis, bilateral filtering, mean shift, total variation, and anisotropic diffusion. The latter is probably the most popular ECPS, for which much research has been carried out in the last fifteen years. However, it is not computationally efficient since it requires much iteration to achieve the desired output. An important aspect of ECPSs is their ability to produce images that are visually similar to paintings. Not all existing ECPSs are suitable for producing such an artistic effect. For this purpose, an interesting class of ECPSs stems from the early work of Kuwahara, where a fast and conceptually simple ECPS is introduced. A symmetric square neighborhood around each pixel is divided in four square windows. The value of the central pixel is replaced by the gray level average over the most homogeneous window, i.e. the window with the lowest standard deviation. Although this operator was not specifically designed for producing artistic images the obtained effects are quite interesting.

Chapter 3

Neural Network

3.1 Introduction

Borrowing from biology, researchers are exploring neural networks—a new, non algorithmic approach to information processing.

A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored within inter neuron connection strengths known as synaptic weights.

Artificial Neural Networks are being counted as the wave of the future in computing. They are indeed self learning mechanisms which don't require the traditional skills of a programmer. But unfortunately, misconceptions have arisen. Writers have hyped that these neuron inspired processors can do almost anything. These exaggerations have created disappointments for some potential users who have tried, and failed, to solve their problems with neural networks. These application builders have often come to the Conclusion that neural network are complicated and confusing. Unfortunately, that confusion has come from the industry itself. An avalanche of articles has appeared touting a large assortment of different neural networks, all with unique claims and specific examples. Currently, only a few of this neuron based structures, paradigms actually, are being used commercially. One particular structure, the feed forward, back propagation network, is by far and away the most popular. Most of the other neural network structures represent models for "thinking" that are still being evolved in the laboratories. Yet, all of these networks are simply tools and as such the only real demand they make is that they require the network architect to learn how to use them. The power and usefulness of artificial neural networks have been demonstrated in several applications including speech synthesis,

diagnostic problems, medicine, business and finance, robotic control, signal processing, computer vision and many other problems that fall under the category of pattern recognition. For some application areas, neural models show promise in achieving human like performance over more traditional artificial intelligence techniques.

3.2 History of Neural Networks

The study of the human brain is thousands of years old. With the advent of modern electronics, it was only natural to try to harness this thinking process. The history of neural networks that was described above can be divided into several periods

First Attempts There were some initial simulations using formal logic. McCulloch and Pitts (1943) developed models of neural networks based on their understanding of neurology. These models made several assumptions about how neurons worked. Their networks were based on simple neurons which were considered to be binary devices with fixed thresholds. The results of their model were simple logic functions such as "a or b" and "a and b". Another attempt was by using computer simulations. Two groups (Farley and Clark, 1954; Rochester, Holland, Haibit and Duda, 1956). The first group (IBM researchers) maintained closed contact with neuroscientists at McGill University. So whenever their models did not work, they consulted the neuroscientists. This interaction established a multidisciplinary trend which continues to the present day.

Promising & Emerging Technology Not only was neuroscience influential the development of neural networks, but psychologists and engineers also contributed to the progress of neural network simulations. Rosenblatt (1958) stirred considerable interest and activity in the field when he designed and developed the Perceptron. The Perceptron had three layers with the middle layer known as the association layer. This system could learn to connect or associate a given input to a random output unit. Another system was the ADALINE (Adaptive Linear Element) which was developed in 1960 by Widrow and Hoff (of Stanford University). The ADALINE was an analogue electronic device made from simple components. The method used for learning was different to that of the Perceptron; it employed the Least Mean Squares (LMS) learning rule.

Period of Frustration & Disrepute In 1969 Minsky and Papert wrote a book in which they generalized the limitations of single layer Perceptrons to multilayered systems. In the book they

said, "...our intuitive judgment that the extension (to multilayer systems) is sterile". The significant result of their book was to eliminate funding for research with neural network simulations. The conclusions supported the disenchantment of researchers in the field. As a result, considerable prejudice against this field was activated.

Innovation Although public interest and available funding were minimal, several researchers continued working to develop neuromorphically based computational methods for problems such as pattern recognition. During this period several paradigms were generated which modern work continues to enhance. Grossberg's (Steve Grossberg and Gail Carpenter in 1988) influence founded a school of thought which explores resonating algorithms. They developed the ART (Adaptive Resonance Theory) networks based on biologically plausible models. Anderson and Kohonen developed associative techniques independent of each other. Klopf (A. Henry Klopf) in 1972 developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called heterostasis. Werbos (Paul Werbos 1974) developed and used the back propagation learning method, however several years passed before this approach was popularized. Back propagation network are probably the most well known and widely applied of the neural today. In essence, the back propagation network is a Perceptron with multiple layers, a different threshold function in the artificial neuron, and a more robust and capable learning rule. Amari (A. Shun Ichi 1967) was involved with theoretical developments, he published a paper which established a mathematical theory for a learning basis (error correction method) dealing with adaptive pattern classification. While Fukushima (F. Kunihiko) developed a step wise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the Cognitron.

Re Emergence: Progress during the late 1970s and early 1980s was important to the re-emergence on interest in the neural network field. Several factors influenced this movement. For example, comprehensive books and conferences provided a forum for people in diverse fields with specialized technical languages, and the response to conferences and publications was quite positive. The news media picked up on the increased activity and tutorials helped disseminate the technology. Academic programs appeared and courses were introduced at most major Universities (in US and Europe). Attention is now focused on funding levels throughout Europe, Japan and the US and as this funding becomes available, several new commercial with applications in industry and financial institutions are emerging.

Today significant progress has been made in the field of neural networks enough to attract a great deal of attention and fund further research. Advancement beyond current commercial applications appears to be possible, and research is advancing the field on many fronts. Neurally based chips are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology.

3.3 Advantages of Neural Networks

Either humans or other computer techniques can use neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, to extract patterns and detect trends that are too complex to be noticed. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. Advantages include

Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

Self Organization: An ANN can create its own organization or representation of the information it receives during learning time.

Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

3.4 Neural Networks versus Conventional Computers

Neural networks take a different approach to problem solving than that of conventional computers.

Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks on the other hand, process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements

(neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task..

The disadvantage of **neural networks** is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand**, conventional computers** use a cognitive approach to problem solving, the way the problem is to solve must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable, if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

3.5 Human and Artificial Neurons Investigating The Similarities

3.5.1 Learning Process in Human Brain

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites.

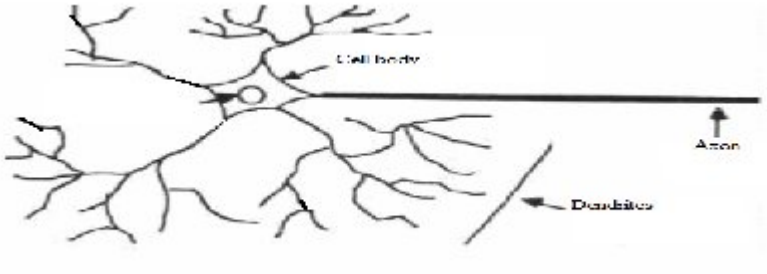


Figure 3.1 Components of Neuron

The neuron sends out spikes of electrical activity through a long, thin stand known as an axon, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite the activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared

with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

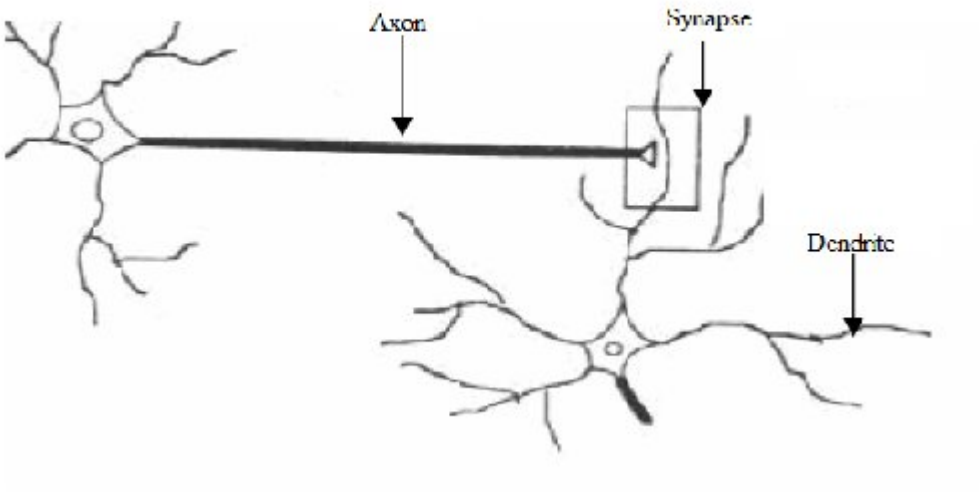


Figure 3.2 The Synapse

3.5.2 Human Neurons to Artificial Neurons

We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections. We then typically program a computer to simulate these features. However because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealizations of real networks of neurons.

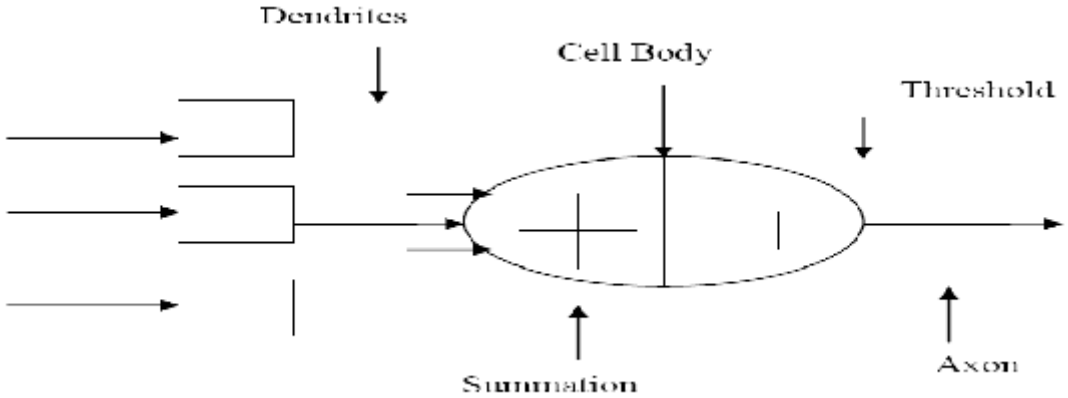


Figure 3.3 Neural Network

3.6 An Engineering Approach

3.6.1 A Simple Neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation, the training mode and the using mode.

- In the training mode, the neuron can be trained to fire (or not), for particular input pattern.
- In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

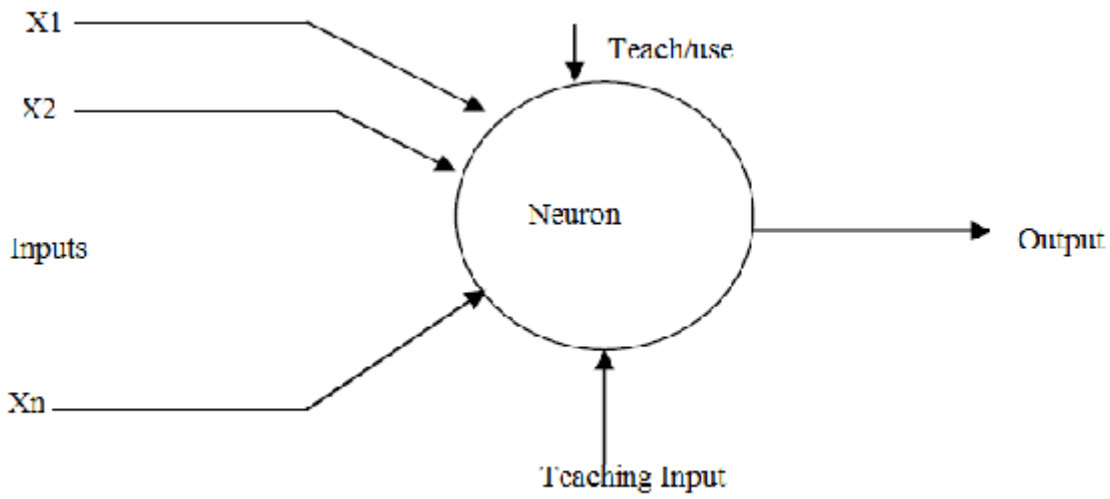


Figure 3.4 A Simple Neuron

3.6.2 Firing Rules

The firing rule is an important concept in neural networks and accounts for their high flexibility. A firing rule determines how one calculates whether a neuron should fire for any input pattern. It relates to all the input patterns, not only the ones on which the node was trained. A simple firing rule can be implemented by using Hamming distance technique. The rule goes as follows

Take a collection of training patterns for a node, some of which cause it to fire (the 1 taught set of patterns) and others, which prevent it from doing so (the 0 taught set). Then the patterns not in

the collection cause the node to fire if, on comparison, they have more input elements in common with the 'nearest' pattern in the 1 taught set than with the 'nearest' pattern in the 0 taught set. If there is a tie, then the pattern remains in the undefined state.

For example, a 3 input neuron is taught to output 1 when the input (X1, X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001. Then, before applying the firing rule, the truth table is

Table 3.1 Truth table before applying the firing rule

X1	0	0	0	0	1	1	1	1
X2	0	0	1	1	0	0	1	1
X3	0	1	0	1	0	1	0	1
X4	0	0	0/1	0/1	0/1	1	0/1	1

As an example of the way the firing rule is applied, take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the 'nearest' pattern is 000, which belongs, in the 0 taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 is equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1).

By applying the firing in every column the following truth table is obtained

Table 3.2 Truth Table after applying the firing rule

X1		0	0	0	0	1	1	1	1
X2		0	0	1	1	0	0	1	1
X3		0	1	0	1	0	1	0	1
OUT		0	0	0	0/1	0/1	1	1	1

.

The difference between the two truth tables is called the generalization of the neuron. Therefore the firing rule gives the neuron a sense of similarity and enables it to respond 'sensibly' to patterns not seen during training

3.6.3 A More Complicated Neuron

The previous neuron doesn't do anything that conventional computers don't do already. A more sophisticated neuron is the McCulloch and Pitts model (MCP). The difference from the previous

model is that the inputs are 'weighted,' the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre set threshold value, the neuron fires. In any other case the neuron does not fire.

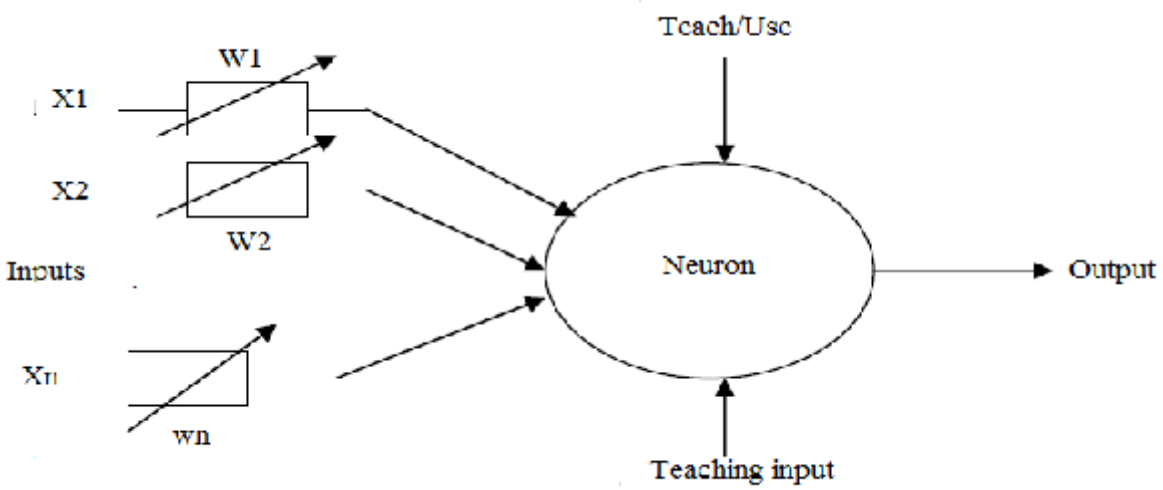


Figure 3.5 MCP Neuron

In mathematical terms, the neuron fires if and only if;

$$X_1W_1 + X_2W_2 + X_3W_3 + ... > T \tag{3.1}$$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold.

3.7 ARCHITECTURE OF NEURAL NETWORKS

3.7.1 NETWORK LAYERS

The common most type of artificial neural network consists of three groups, or layers, of units a layer of "**input**" units is connected to a layer of "hidden" units, which is connected to a layer of "**output**" units.

- The activity of the input units represents the raw information that is fed into the network.

- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units

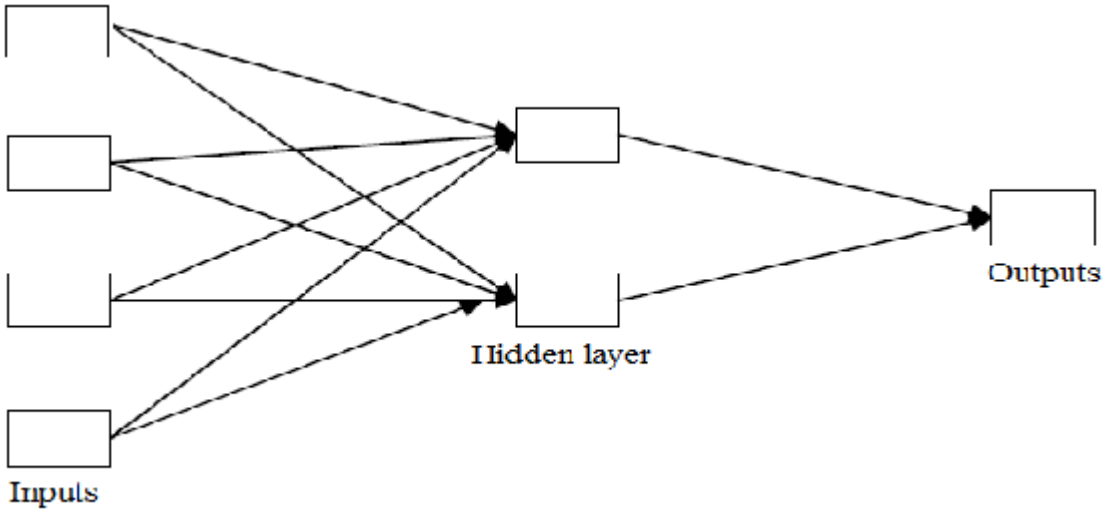


Figure 3.6 Network Layers

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single layer and multi layer architectures. The single layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multilayer organizations. In multi layer networks, layer, instead of following a global numbering, often numbers units.

3.7.2 FEED FORWARD NETWORKS

Feed forward Ann's allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed forward Ann's tend to be straightforward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom up or top down.

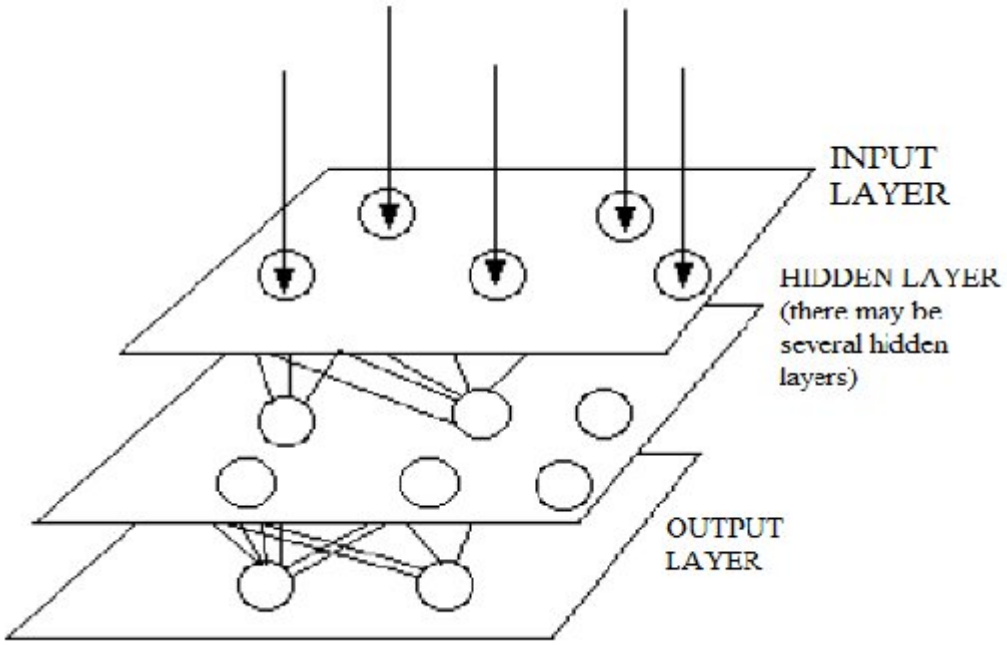


Figure 3.7 Feed Forward Neural Network

3.7.3 Feedback Networks

Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic, their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single layer organizations.

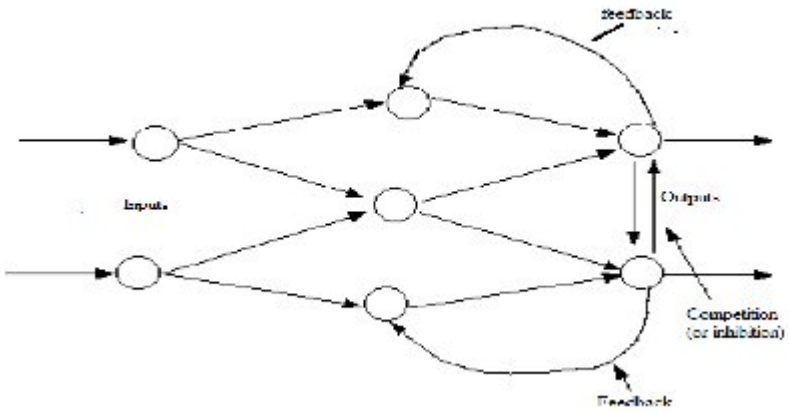


Figure 3.8 Feedback Neural Network

3.7.4 Perceptrons

The Perceptron calculates a weighted sum of inputs and compares it to a threshold. If the sum is higher than the threshold, the output is set to 1, otherwise to -1, depending upon activation function.

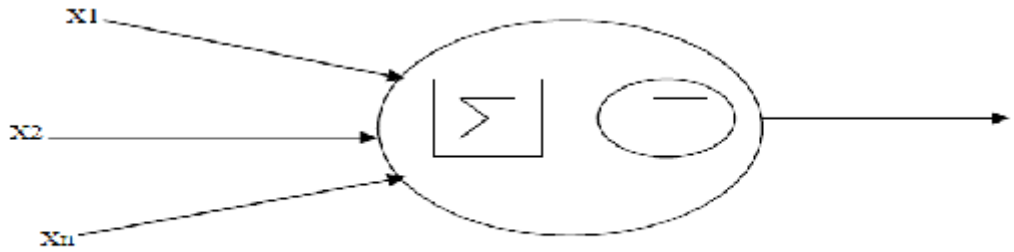


Figure 3.9 The Mcculloh Pits Model

In 1969 Minsky and Papert wrote a book in which they described the limitations of single layer Perceptrons. The impact that the book had was tremendous and caused a lot of neural network researchers to loose their interest. The book was very well written and showed mathematically that single layer Percetrons could not do some basic pattern recognition operations like determining the parity of a shape or determining whether a shape is connected or not. What they did not realize, until the 80's, is that given the appropriate training, multilevel Perceptrons can do these operations.

3.8 Transfer Function

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input output function (transfer function) that is specified for the units. This function typically falls into one of three categories

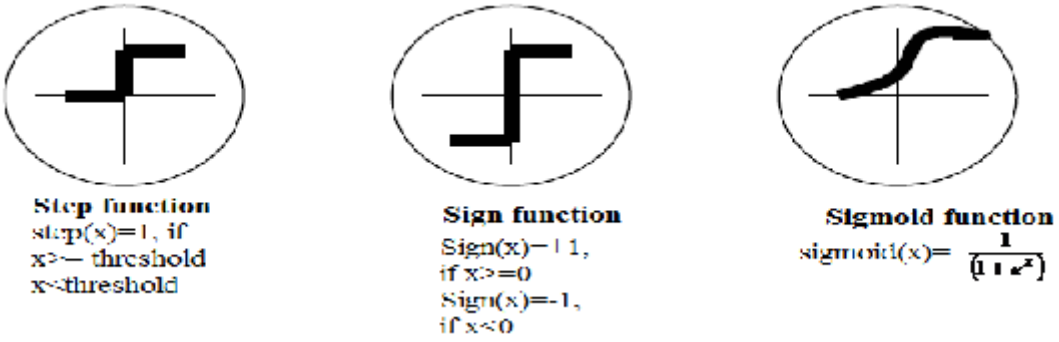


Figure 3.10 Transfer Function

Linear (or ramp): the output activity is proportional to the total weighted output.

Threshold: the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

Sigmoid: the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another, and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence. A three layer network can be taught to perform a particular task by using the following procedure:

- The network is presented with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
- It is determined how closely the actual output of the network matches the desired output.
- The weight of each connection can be changed so that the network produces a better approximation of the desired output.

3.9 Learning Algorithms of Neural Networks

Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes takes place. All learning methods used for neural networks can be classified into two major categories.

3.9.1 Supervised Learning which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error correction learning, reinforcement learning and stochastic learning.

An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well known method, which is common to many learning paradigms, is the least mean square (LMS) convergence.

3.9.1.1 The Back Propagation Learning

A back propagation neural network uses a feed forward topology, supervised learning, and back propagation learning algorithm. This algorithm was responsible in large part for the re emergence of neural networks in the mid 1980s. Back propagation is a general purpose learning algorithm. It is powerful but also expensive in terms of computational requirements for training. A back propagation network with a single hidden layer of processing elements can model any continuous function to any degree of accuracy (given enough processing elements in the hidden layer).

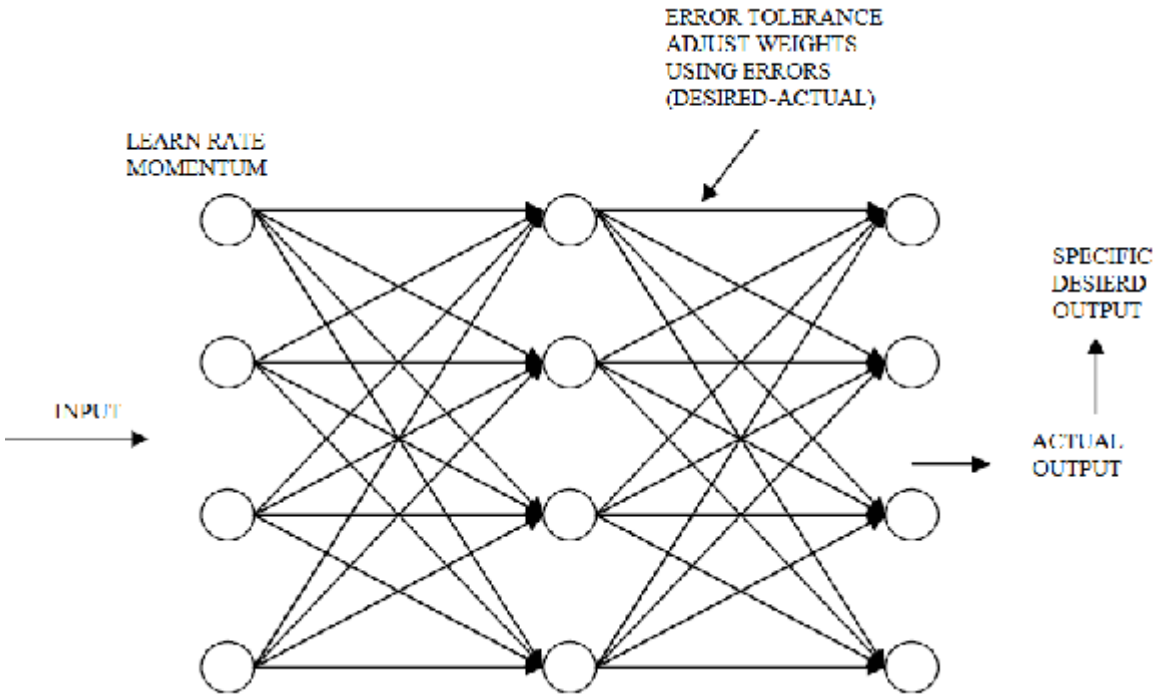


Figure 3.11 Back Propagation network

There are literally hundreds of variations of back propagation in the neural network literature, and all claim to be superior to "basic" back propagation in one way or the other. Indeed, since back propagation is based on a relatively simple form of optimization known as gradient descent, mathematically astute observers soon proposed modifications using more powerful techniques such as conjugate gradient and Newton's methods. However, "basic" back propagation is still the most widely used variant. Its two primary virtues are that it is simple and easy to understand, and it works for a wide range of problems. The basic back propagation algorithm consists of three steps.

- The input pattern is presented to the input layer of the network. These inputs are propagated through the network until they reach the output units. This forward pass produces the actual or predicted output pattern.
- Because back propagation is a supervised learning algorithm, the desired outputs are given as part of the training vector. The actual network outputs are subtracted from the desired outputs and an error signal is produced.
- This error signal is then the basis for the back propagation step, whereby the errors are passed back through the neural network by computing the contribution of each hidden processing unit and deriving the corresponding adjustment needed to produce the correct output. The connection weights are then adjusted and the neural network has just "learned" from an experience.

Two major learning parameters are used to control the training process of a back propagation network. The learn rate is used to specify whether the neural network is going to make major adjustments after each learning trial or if it is only going to make minor adjustments. Momentum is used to control possible oscillations in the weights, which could be caused by alternately signed error signals. While most commercial back propagation tools provide anywhere from 1 to 10 or more parameters for you to set, these two will usually produce the most impact on the neural network training time and performance.

3.9.1.2 Reinforcement Learning

Reinforcement learning is a form of supervised learning because the network gets some feedback from its environment. The feedback signal (yes/no reinforcement signal) is only evaluative, not instructive, i.e. if the reinforcement signal says that a particular output is wrong and it gives no hint as to what the right answer should be. It is therefore important in a reinforcement learning network to implement some source of randomness in the network, so that the space of possible outputs can be explored until a correct value is found. Reinforcement learning is sometimes called "learning with a critic" as opposed to "learning with a teacher" which refers to more traditional learning schemes were an error signal is generated which also contain information in which direction the synaptic weights of the networks should be changed in order to improve the performance. In reinforcement learning problems it is common to think explicitly of a network

functioning in an environment. The environment supplies the inputs to the network, receives its output and then provides the reinforcement signal.

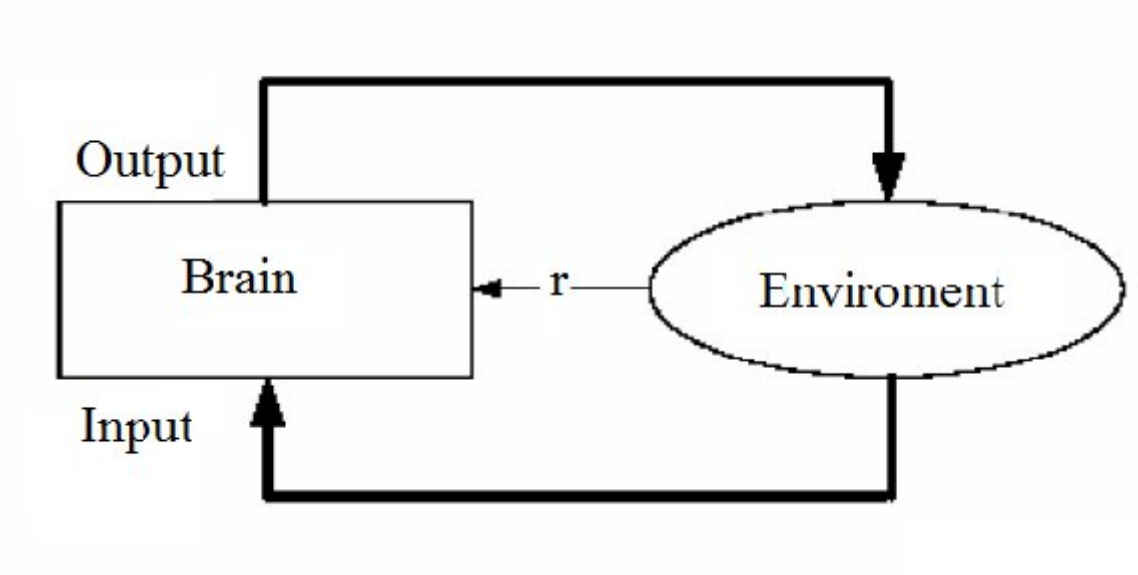


Figure 3.12 The principal layout for a reinforcement learning agent

3.9.2 Unsupervised Learning

3.9.2.1 Hebbian Learning

In the year 1949 Donald Hebb postulated a learning rule that states that the connection between two neurons is strengthened if the neurons fire simultaneously (or within a time interval). The Hebbian learning rule specifies how much the weight between two neurons should be increased or decreased in proportion to their activation. If X_j and X_i are the activations of neurons i and j, W_{ij} is the connection weight between them, and the learning rate, then Hebb's rule in its basic form can be written as

$$\Delta W_{ij} = * X_i * X_j \tag{3.3}$$

where ΔW_{ij} is the change of the connection W_{ij}. Hebbian learning is a type of unsupervised learning that does not consider the result of the output, i.e. it is a correlation based form of learning. We may expand and rephrase it as a two part rule

- If two neurons on either side of a synapse are activated simultaneously, then the strength of that synapse is selectively increased.

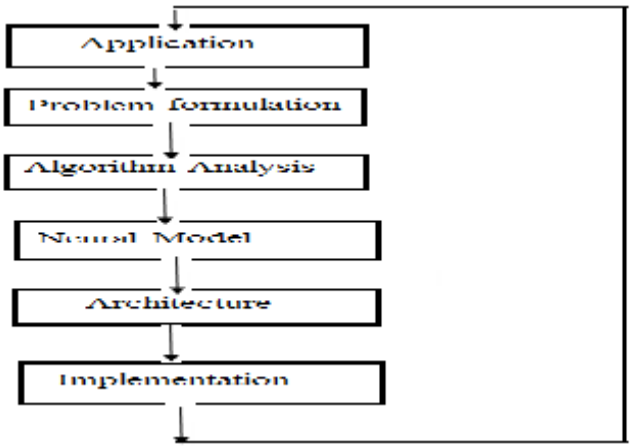
- If two neurons on either side of synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.

3.9.2.2 Competitive Learning

In competitive learning, the output neurons of a neural network compete among themselves to become active (fired) whereas in a neural network based on hebbian learning several output neurons may be active simultaneously, in competitive learning only a single output neuron is active at only one time. There are three basic elements to a competitive learning rule

- A set of neurons that are all same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.
- A limit imposed on the strength of each neuron.
- A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, or only one neuron per group is active at a time. The neuron that wins the competition is called a winner takes all neuron.

3.10 Applications of Neural Network



,

Figure 3.13 Flowchart For Implementation of Application

In order to have an integrated understanding on neural networks we adopt the next perspective, called top down, from application, algorithm to architecture. The approach is application motivated, theoretically based, and implementation oriented. The main applications are for signal processing and pattern recognition. The algorithmic treatment represents a combination of mathematical theory and heuristic justification for neural models. The ultimate objective is the

implementation of digital neuro computers, embracing technologies of VLSI, adaptive, digital and parallel processing

3.10.1 Neural Networks in Practice

Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries. Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including

- Sales Forecasting
- Industrial Process Control
- Customer Research
- Data Validation
- Risk Management
- Target Marketing

But to give you some more specific examples, ANN are also used in the following specific paradigms recognition of speakers in communications, diagnosis of hepatitis, recovery of telecommunications from faulty software, interpretation of multi meaning Chinese words, undersea mine detection, texture analysis; three dimensional object recognition, hand written word recognition, and facial recognition.

3.10.2 Neural Networks in Medicine

Artificial Neural Networks (ANN) is currently a 'hot' research area in medicine and it is believed that they will receive extensive application to biomedical systems in the next few years. At the moment, the research is mostly on modeling parts of the human body and recognizing diseases from various scans (e.g. cardiograms, CAT scans, ultrasonic scans, etc.).

Neural networks are ideal in recognizing diseases using scans since there is no need to provide a specific algorithm on how to identify the disease. Neural networks learn by example so the details of how to recognize the disease are not needed. What is needed is a set of examples that are representative of all the variations of the disease. The quantity of examples is not as important as the 'quantity'. The examples need to be selected very carefully if the system is to perform reliably and efficiently.

3.10.2.1 Modeling and Diagnosing The Cardiovascular System

Neural Networks are used experimentally to model the human cardiovascular system. Diagnosis can be achieved by building a model of the cardiovascular system of an individual and comparing it with the real time physiological measurements taken from the patient. If this routine is carried out regularly, potential harmful medical conditions can be detected at an early stage and thus make the process of combating the disease much easier.

A model of an individual's cardiovascular system must mimic the relationship among physiological variables (i.e., heart rate, systolic and diastolic blood pressures, and breathing rate) at different physical activity levels. If a model is adapted to an individual then it becomes a model of the physical condition of that individual. The simulator will have to be able to adapt to the features of any individual without the supervision of an expert. This calls for a neural network.

Another reason that justifies the use of ANN technology is the ability of Ann's to provide sensor fusion which is the combining of values from several different sensors. Sensor fusion enables the Ann's to learn complex relationships among the individual sensor values, which would otherwise be lost if the values were individually analyzed. In medical modeling and diagnosis, this implies that even though each sensor in a set may be sensitive only to a specific physiological variable, Ann's are capable of detecting complex medical conditions by fusing the data from the individual biomedical sensors.

3.10.2.2 Electronic Noses

Ann's are used experimentally to implement electronic noses. Electronic noses have several potential applications in telemedicine. Telemedicine is the practice of medicine over long distances via a communication link. The electronic nose would identify odors in the remote surgical environment. These identified odors would then be electronically transmitted to another site where a door generation system would recreate them. Because the sense of smell can be an important sense to the surgeon, telesmell would enhance telepresent surgery.

3.10.2.3 Instant Physician

An application developed in the mid 1980s called the "instant physician" trained an auto associative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis, and treatment for a particular case. After training,

the net can be presented with input consisting of a set of symptoms; it will then find the full stored pattern that represents the "best" diagnosis and treatment.

3.10.3 Neural Networks in Business

Business is a diverted field with several general areas of specializations such as accounting or financial analysis. Almost any neural network application would fit into one business area or financial analysis. There is some potential for using neural networks for business purposes, including resource allocation and scheduling. There is also a strong potential for using neural networks for database mining that is, searching for patterns implicit within the explicitly stored information in databases. Most of the funded work in this area is classified as proprietary. Thus, it is not possible to report on the full extent of the work going on. Most work is applying neural networks, such as the Hopfield Tank network for optimization and scheduling.

Chapter 4
Introduction to Back Propagation

4.1 INTRODUCTION

Back propagation is a form of supervised learning for multi layer nets, also known as the generalized delta rule. Error data at the output layer is back propagated to earlier ones, allowing incoming weights to these layers to be updated. It is most often used as training algorithm in current neural network applications. The back propagation algorithm was developed by Paul Werbos in 1974 and rediscovered independently by Rumelhart and Parker. Since its rediscovery, he back propagation algorithm has been widely used as a learning algorithm in feed forward multilayer neural networks.

4.2 History of Algorithm

Minsky and Papert (1969) showed that there are many simple problems such as the exclusive or problem, which linear neural networks cannot solve. Note that term "solve" means learn the desired associative links. Argument is that if such networks cannot solve such simple problems how could they solve complex problems in vision, language, and motor control. Solutions to this problem were

- Select appropriate "recoding" scheme which transforms inputs
- Perceptron Learning Rule Requires that you correctly "guess" an acceptable input to hidden unit mapping.
- Back propagation learning rule Learn both sets of weights simultaneously.

4.3 Learning With Back Propagation Algorithm

The back propagation algorithm is an involved mathematical tool, however, execution of the training equations is based on iterative processes, and thus is easily implement able on a computer.

- Weight changes for hidden to output weights just like Widrow Hoff learning rule.

- Weight changes for input to hidden weights just like Widrow Hoff learning rule but error signal is obtained by "back propagating" error from the output units.

During the training session of the network, a pair of patterns is presented (X_k, T_k), where X_k in the input pattern and T_k is the target or desired pattern. The X_k pattern causes output responses at teach neurons in each layer and, hence, an output O_k at the output layer. At the output layer, the difference between the actual and target outputs yields an error signal. This error signal depends on the values of the weights of the neurons I each layer. This error is minimized, and during this process new values for the weights are obtained. The speed and accuracy of the learning process that is, the process of updating the weights also depends on a factor, known as the learning rate. Before starting the back propagation learning process, we need the following

- The set of training patterns, input, and target
- A value for the learning rate
- A criterion that terminates the algorithm
- A methodology for updating weights
- The nonlinearity function (usually the sigmoid)
- Initial weight values (typically small random values)

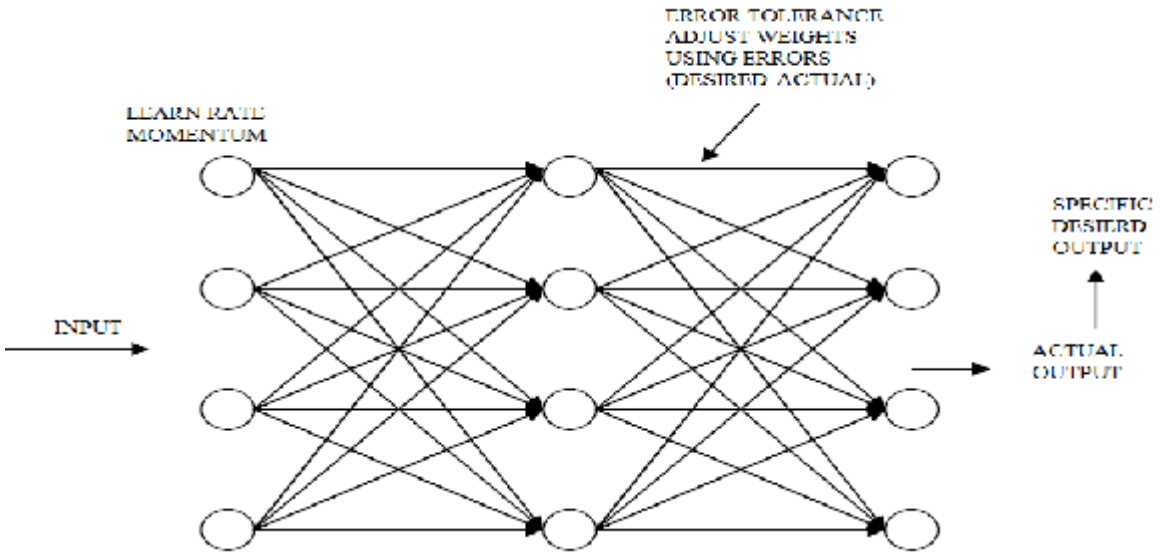


Figure 4.1 Back Propagation Network

The process then starts by applying the first input pattern X_k and the corresponding target output T_k. The input causes a response to the neurons of the first layer, which in turn cause a response to the neurons of the next layer, and so on, until a response is obtained at the output layer. That response is then compared with the target response; and the difference (the error signal) is calculated. From the error difference at the output neurons, the algorithm computes the rate at which the error changes as the activity level of the neuron changes. So far, the calculations were computed forward (i.e., from the input layer to the output layer). Now, the algorithm steps back one layer before that output layer and recalculate the weights of the output layer (the weights between the last hidden layer and the neurons of the output layer) so that the output error is minimized. The algorithm next calculates the error output at the last hidden layer and computes new values for its weights (the weights between the last and next to last hidden layers). The algorithm continues calculating the error and computing new weight values, moving layer by layer backward, toward the input.

When the input is reached and the weights do not change, (i.e., when they have reached a steady state), then the algorithm selects the next pair of input target patterns and repeats the process. Although responses move in a forward direction, weights are calculated by moving backward, hence the name back propagation.

4.4 Implementation of Back Propagation Algorithm

The back propagation algorithm consists of the following steps

- Each Input is then multiplied by a weight that would either inhibit the input or excite the input. The weighted sum of then inputs in then calculated

First, it computes the total weighted input X_j, using the formula

$$X_j = \sum_i Y_i W_{ij} \tag{4.1}$$

Where Y_i is the activity level of the j^{th} unit in the previous layer and W_{ij} is the weight of the connection between the i^{th} and the j^{th} unit.

Then the weighed X_j is passed through a sigmoid function that would scale the output in between 0 and 1.

- Next, the unit calculates the activity yj using some function of the total weighted input. Typically we use the sigmoid function

$$Y_j = \frac{1}{1+e^{-x_j}} \tag{4.2}$$

Once the output is calculated, it is compared with the required output and the total Error E is computed.

- Once the activities of all output units have been determined, the network computes the error E, which is defined by the expression

$$E = \frac{1}{2}\sum_j \left(Y_j - d_j\right)^2 \tag{4.3}$$

where Y_j is the activity level of the i^{th} unit in the top layer and d_j is the desired output of the i^{th} unit.

Now the error is propagated backwards.

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA) is the difference between the actual and the desired activity.

$$EA_j = \frac{\delta E}{\delta Y_j} = Y_j - d_j \tag{4.4}$$

2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$EI_J = \frac{\delta E}{\delta x_j} = \frac{\delta E}{\delta Y_j} \times \frac{\delta Y_j}{\delta x_j} = EA_j Y_j \left(1 - y_j\right) \tag{4.5}$$

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$EW_{ij} = \frac{\delta E}{\delta W_{ij}} = \frac{\delta E}{\delta x_j} \times \frac{\delta x_j}{\delta W_{ij}} = EI_j Y_i \tag{4.6}$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. This crucial step allows back propagation to be applied to multi layer networks. When the

activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these separate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output unit. By using steps 2 and 4, we can convert the EA's of one layer of units into EA's for the previous layer. This procedure can be repeated to get the EA's for as many previous layers as desired. Once we know the EA of a unit, we can use steps 2 and 3 to compute the EW's on its incoming connections.

$$EA_i = \frac{\delta E}{\delta Y_i} = \sum_j \frac{\delta E}{\delta x_j} \times \frac{\delta x_j}{\delta Y_i} = \sum_j EI_j W_{ij} \tag{4.7}$$

4.5 DRAWBACKS

Although widely used, the back propagation algorithm has not escaped criticism. The method of backwards calculating weights does not seem to be biologically plausible; neurons do not seem to work backward to adjust the efficacy of their synaptic weights. Thus, the back propagation learning algorithm is not viewed by many as a learning process that emulates the biological world but as a method to design a network with learning.

Second, the algorithm uses a digital computer to calculate weights. When the final network is implemented in hardware, however, it has lost its plasticity. This loss is in contrast with the initial motivation to develop neural networks that emulate brain like networks and are adaptable (plastic) enough to learn new patterns. If changes are necessary, a computer calculates anew the weight values and updates them. This means that the neural network implementation still depends on a digital computer.

The algorithm suffers from extensive calculations and, hence, slows training speed. The time required to calculate the error derivatives and to update the weights on a given training exemplar is proportional to the size of the network. The amount of computation is proportional to the number of weights. In large networks, increasing the number of training patterns causes the learning time to increase faster that the network. The computational speed inefficiency of this algorithm has triggered an effort to explore techniques that accelerated the learning time by at

least a factor of 2. Even these accelerated techniques, however, do not make the back propagation learning algorithm suitable in many real time applications.

Despite it wide applicability, the error back propagation algorithm cannot be applied to all neural network systems which can be imagined. In particular, the algorithm requires that the activation functions of each of the neurons in the network be both continuous and differentiable. Several historically important neural network architectures use activation functions which do not satisfy this condition these include the discontinuous linear threshold activation function of the original perceptron of Rosenblatt and the continuous but non differentiable linear ramp activation function of the units in the brain state in ebox model of Anderson et al.

CHAPTER 5

Research Methodologies

5.1 Matlab Software

Short for "matrix laboratory", MATLAB is a numerical computing environment and programming language, Created by The MathWorks. MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. MATLAB is built around the MATLAB language, sometimes called M code or simply M [http matlab].

5.2 Image Processing Toolbox

Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image processing operations. In this work, we have used image processing toolbox of MATLAB version 7.5.

5.3 Algorithm

In this study, an attempt has been made to train the network for different epoch value of leena, girl and cameraman image. Based on different training epoch, distinguish compression ratio, peak signal to noise ratio, bits per pixel are obtained. Following steps have been performed to achieve this objective.

Step 1 The image is read by using the command imread. This is an inbuilt command in MATLAB processing toolbox.

Syntax of image read function is I = imread (filename, fmt)

Definition imread reads a grayscale or color image from the file specified by the string filename, the string fmt specifies the format of the file. I is a matrix each element of I is a pixel with the corresponding gray scale of the image

Step 2

Resize the loaded image to a standard size of 512×512. The images taken for testification have a lot of variation in their sizes and hence cannot be compared on the same basis. For large sized images, such as 1024×1024, the computation time for compression is found to be more. And if the image size is taken smaller than 256×256, then the useful data is liable to get lost.

Step 3 Compression

A three layer feed forward network is shown in figure 5.1, the network is called fully connected. Because there are all to all connections between two adjacent neuron layers. The number of neurons (also called units) in each layer is N_i, N_h and N_o for the input, hidden, and output neuron layers, respectively. The network can be extended to any number of layers; however, because most applications use two weight layers, the description here has been restricted to two layer networks. The BP learning phase for a pattern consists of a forward phase followed by a backward phase. The training algorithm of back propagation involves four stages is as follows

- Initialization of weights
- Feed forward
- Back propagation of errors
- Updation of the weights and biases

There are two types of learning in back propagation sequential learning and batch learning. In sequential learning a given input pattern is propagated forward, the error is determined and back propagated, and the weights are updated. In batch learning the weights are updated only after the entire set of training network has been presented to the network. Thus the weights update is only performed after every epoch. To train then et work, the proposed training algorithm used in the back propagation algorithm is given below.

The main steps are as follows

1. Initialize the weights to small random values.

2. Select a training vector pair (input and the corresponding output) from the training set and present the input vector to the inputs of the network.

3. Calculate the actual outputs this is the forward phase.

4. According to the difference between actual and desired outputs (error). Adjust the weights W_o and W_h to reduce the difference this is the backward phase.

5. Repeat from step 2 for all training vectors.

6. Repeat from step 2 until the error is acceptably small

Back Propagation learning algorithm. In the forward phase the hidden layer weight matrix W_h is multiplied by the input vector X=(X1, X2, X3…….X n) to calculate theα

hidden layer output

$$Y_{h,j} = \text{f}(\sum W_{hji,} * X)$$

is a demand of the training algorithm Where $W_{h,ji}$ is the weight connecting input unit I to unit j in the hidden neuron layer. The function f is a nonlinear activation function. Normally the S shaped sigmoid function F (α) = $1/1+e^{-\alpha}$ is used. It compresses the output value to lie in (0, 1), as shown in figure 5.2. Moreover the function is differentiable

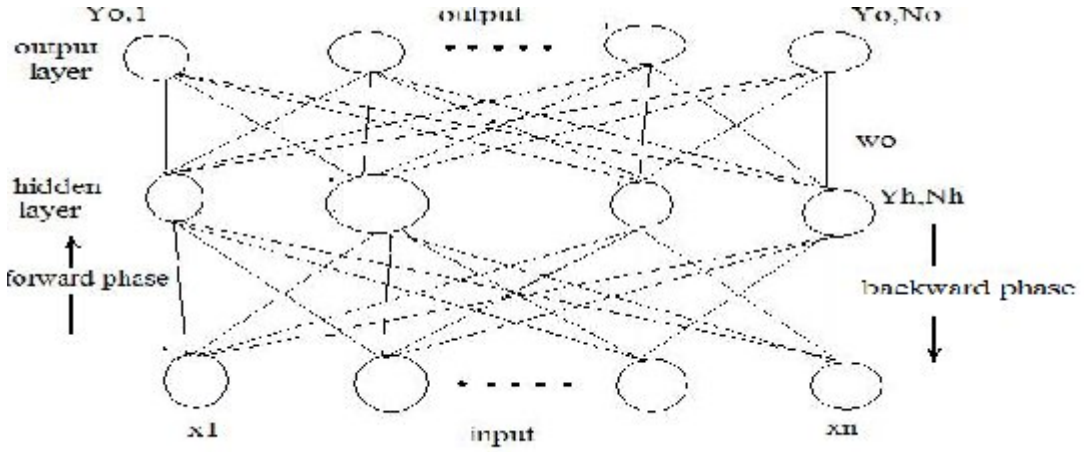


Figure 5.1 A Three Weight Layer Feed Forward Neural Network

.

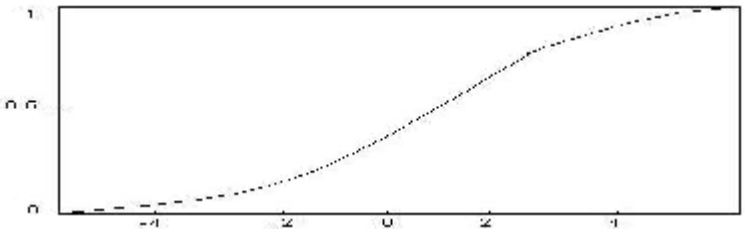


Figure 5.2 The Sigmoid Function f (α) =$1/1+e^{-\alpha}$

The output from the hidden layer $Y_{h,j}$ is used to calculate the output of the network $Y_{o,k}$

$$Y_{o,k} = f(\sum W_{o,kj} * Y)_{h,j}$$

The error measure E_p for a training pattern p is given by

$$E_p = 1/2 \sum (d_{p,k} - Y_{p,k,o})^2$$

The overall error measure for a training set of P patterns is

$$E = \sum E_P$$

In the following expressions, the pattern index p has been omitted on all variables to improve clarity. In the backward phase the target, d, and output, Y_O, are compared and the difference (error) is used to adapt the weights to reduce the error.

The error used to update the weights can be shown to be

$$\delta_{o,k} = Y_{o,k}(1 - Y_{o,k})(d_k - Y_{o,k})$$

Similar to computing the output delta error, the hidden delta error value for neuron j is

$$\delta_{h,j} = Y_{h,j}(1 - Y_{h,j}) \sum \delta_{o,k} W_{o,kj}$$

The error is not explicitly given and is computed based on the impact of the fan in of the output delta errors. To perform steepest descent in the weight space, the weight changes become

$$\Delta W_{o,kj} = \eta \delta_{o,k} Y_{h,j}$$
$$\Delta W_{h,ji} = \eta \delta_{h,i} X_i$$

Where η is the learning rate coefficient

If learning by pattern is applied, the output layer weights are changed to $W_{o,kj}$

$$W_{o,kj} = W_{o,kj} + \eta \ \ \delta_{o,k} * Y_{h,j}$$

the hidden layer weights are updated accordingly

$$W_{h,ji} = W_{h,ji} + \delta \eta_{h,i} * X_i$$

The training continues for each vector in the training set until the error for the entire set becomes acceptably small

Training Algorithms

The training process requires a set of examples of proper network behavior network inputs p and target outputs t. In previous chapters it has been already mentioned that the back propagation neural network is used for behavior classification. In this work neural network is constructed with 2 units in the input layer and one unit in the output layer. Only one hidden layer is used in

this work and there is option to choose different number of hidden nodes for the system. This can be simply done by changing the value of the variable for hidden units in the implementation. . A weight value is associated with each of the connections. The output of the neural network will be our desired target output.

Different test runs were made for each of the following training algorithm and also tried with different number of hidden nodes. The training function that has been used in this work is given in the table 5.2 with some other training functions

Table 5.2 Description of Different Neural Network Training Functions

Function	**Description**
Trainbfg	BFGS quasi Newton method. Requires storage of approximate Hessian matrix and has more computation in each iteration than conjugate gradient algorithms, but usually converges in less iteration.
Trainoss	One step secant method. Compromise between conjugate gradient methods and quasi Newton methods.
Traingda	Levenberg Marquardt algorithm. Fastest training algorithm for networks of moderate size. Has memory reduction feature for use when the training set is large.

Step 5 Testing

The system is already trained with the normal given data. In the training phase the input pattern and output pattern is given to the network but in this phase only the input pattern is given. After initialization of the weights, the input units of input layer are activated with the input patterns that have been taken from the input file. The outputs of the input layer are propagated towards the output layer similarly as training. The calculated output from the output unit of output layer is considered as the desired output pattern. This output will show us whether the given input pattern represents the desired output has come or not

Step7

The parameters, PSNR (peak signal to noise ratio), RMSE (root mean square error), compression ratio, bits per pixel are calculated.

$$\text{RMSE} = \left[\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\hat{f}(i,j) - f(i,j) \right]^2 \right]^{1/2}$$

$$\text{PSNR} = 10 \log_{10} \left[\frac{M \times N}{RMSE^2} \right]$$

Where M*N is the size of the image, f(i, j) and f(i,j) are the matrix element of the decompressed and the original image at (i.j) pixel. In order to evaluate the performance of image compression system, compression ratio matrix is often employed.

Bits per pixel

```
Lista = dir(file_name)
Mybytes=  lista.bytes
Bitsperpixel= (mybytes*8)/(num(I0))
Disp('bits per pixel')
```

Bits per pixel is a computer graphics term describing the number of bits used to represent the color of a single pixel in a bitmapped image or video frame buffer. This concept is also known as color depth, particularly when specified along with the number of bits used. Higher color depth gives a broader range of distinct colors Color depth is only one aspect of color representation expressing how finely levels of color can be expressed, the other aspect is how broad a range of colors can be expressed.

Chapter 6

Simulation and Testing

6.1 Introduction

The quality of compressed image can be measured by many parameters, which compare to the different compression technique. The most commonly used parameters are Root Mean Square error (RMSE), peak signal to noise ratio error (PSNR), compression ratio(CR) .The PSNR value used to measure the difference between a decoded image and its original image as follows. In general, the larger the PSNR value, the better will be the decoded image quality

$$\text{RMSE} = \left[\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\hat{f}(i,j) - f(i,j) \right]^2 \right]^{1/2} \tag{6.1}$$

$$\text{PSNR} = 10 \log_{10} \left[\frac{M \times N}{RMSE^2} \right] \tag{6.2}$$

Where M*N is the size of the image, $\hat{f}$(i, j) and f(i, j) are the matrix element of the decompressed and the original image at (i ,j) pixel. In order to evaluate the performance of image compression system, compression ratio matrix is often employed. In our results, compression ratio (CR) is computed as the ratio of non zero entries in the original image to the non zero entries in the decompressed image.

CR = original image /compressed image size

CR%= (1 (1/CR))*100 (6.3)

Image compression using Neural Network is conducted on many images. The first image is standard Lena Image, whose faced has graced the pages of many image processing books, paper and project .The second and third images are Girl and Cameraman.

6.2 Testing of Image 1

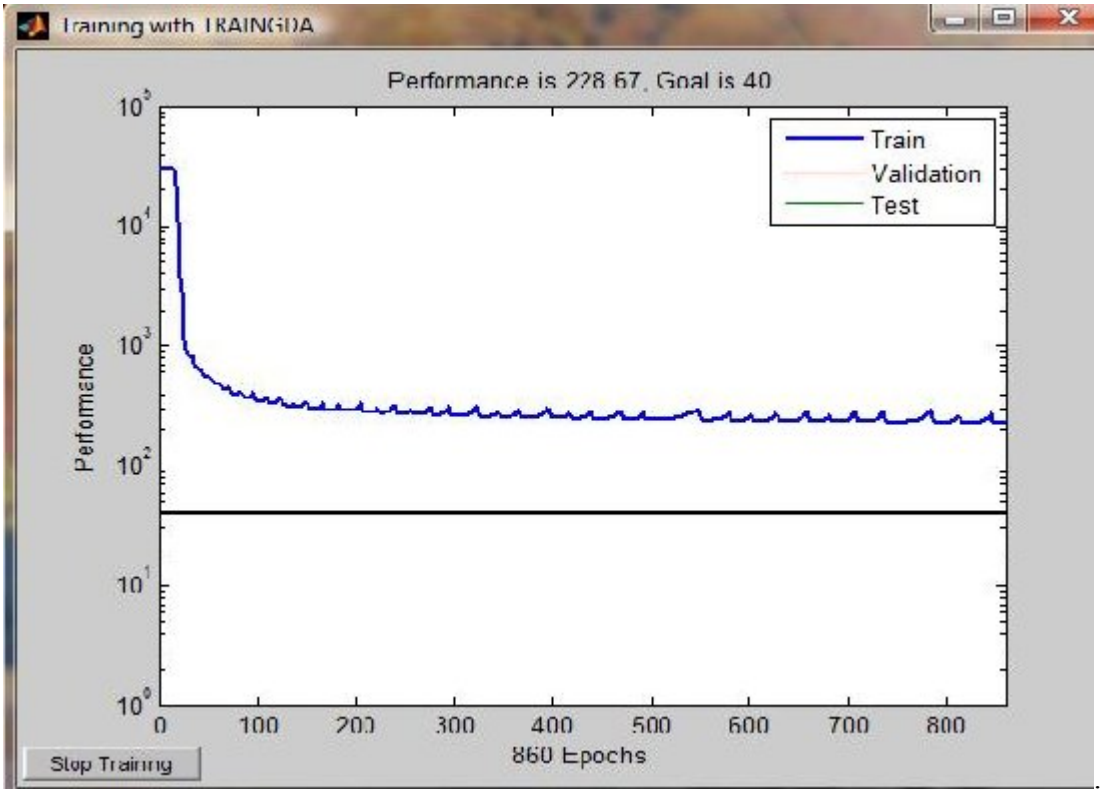


Figure 6.1 Output of training dataset using Traingda Training Function

The graph shown in figure 6.1 represents the output of the training of the network and 860 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network is 228.67.

Figure 6.2 Original and Decompressed Image of Lena after 860 epochs

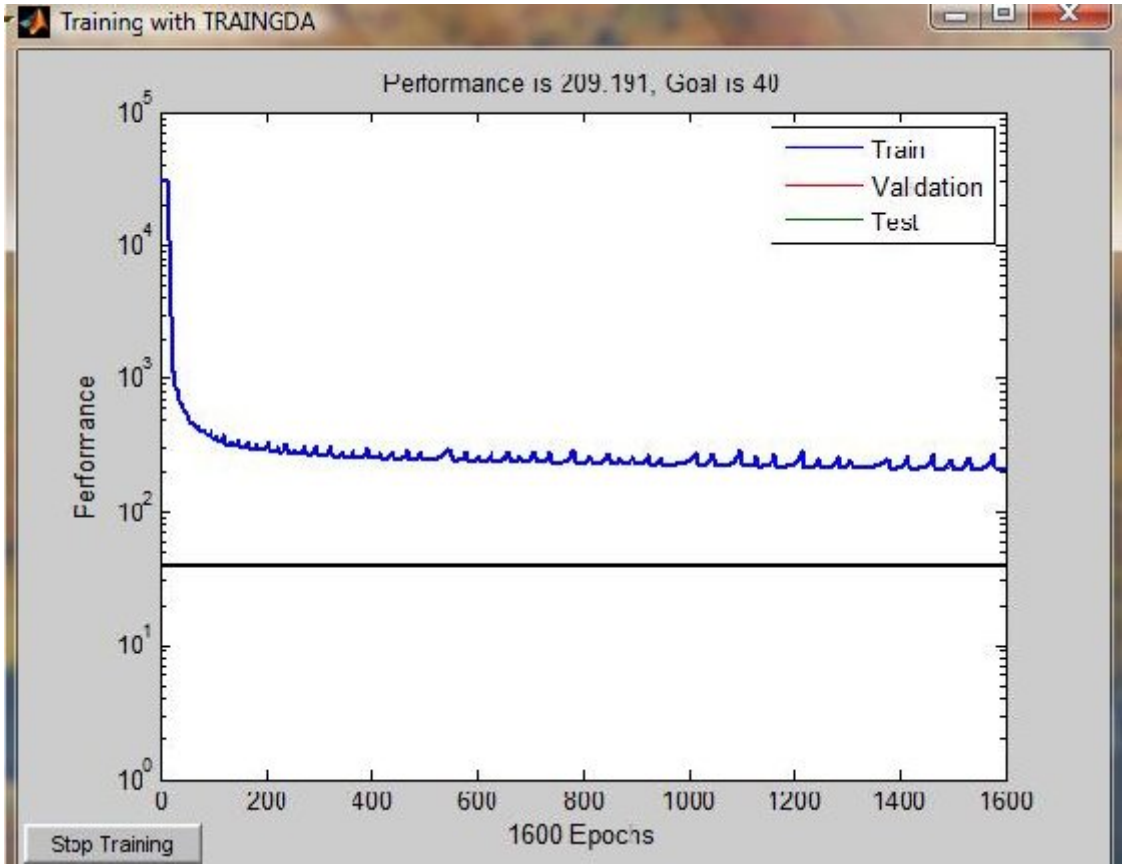


Figure 6.3 Output of training dataset using Traingda Training Function

The graph shown in figure 6.3 represents the output of the training of the network and 1600 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been taken upto 209.191.

Figure 6.4 Original and Decompressed Image of Lena after 1600 epochs

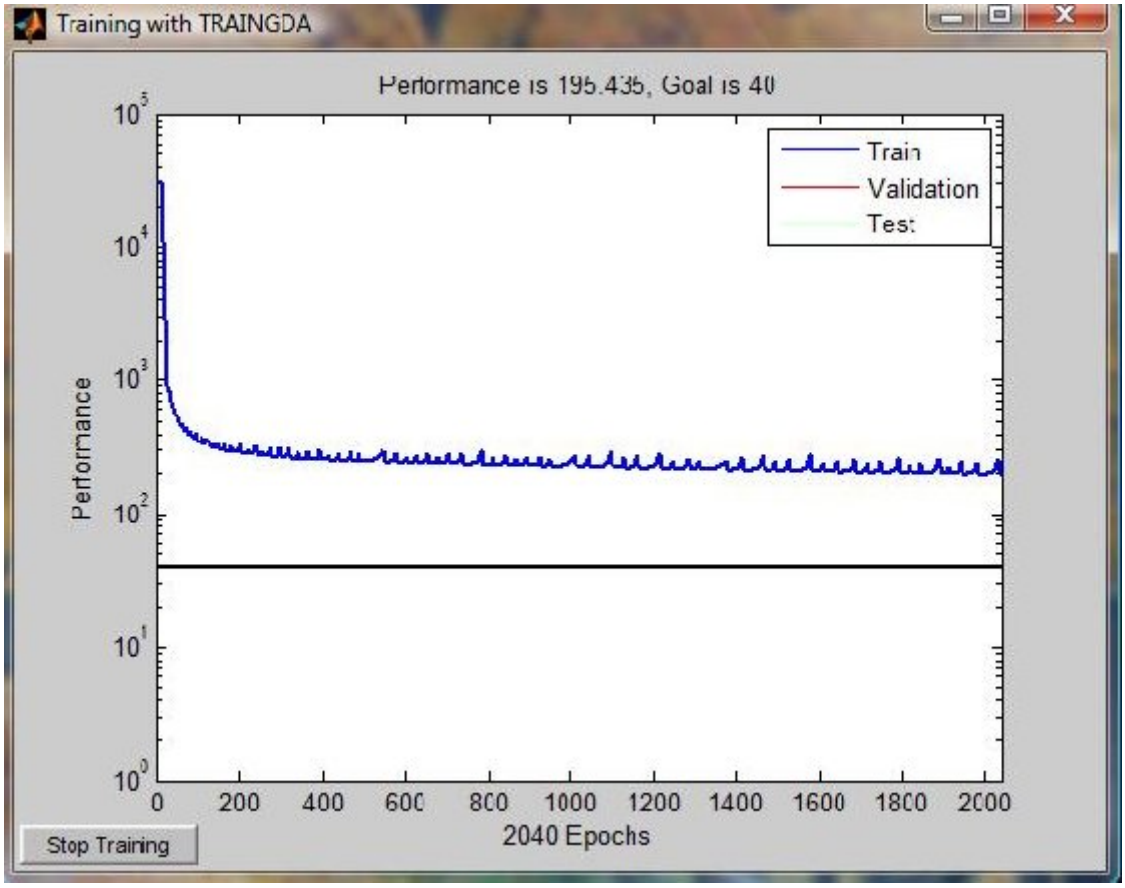


Figure 6.5 Output of training dataset using Traingda Training Function

The graph shown in figure 6.3 represents the output of the training of the network and 2040 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 195.435.

Figure 6.6 Original and Decompressed Image of Lena after 2040 epochs

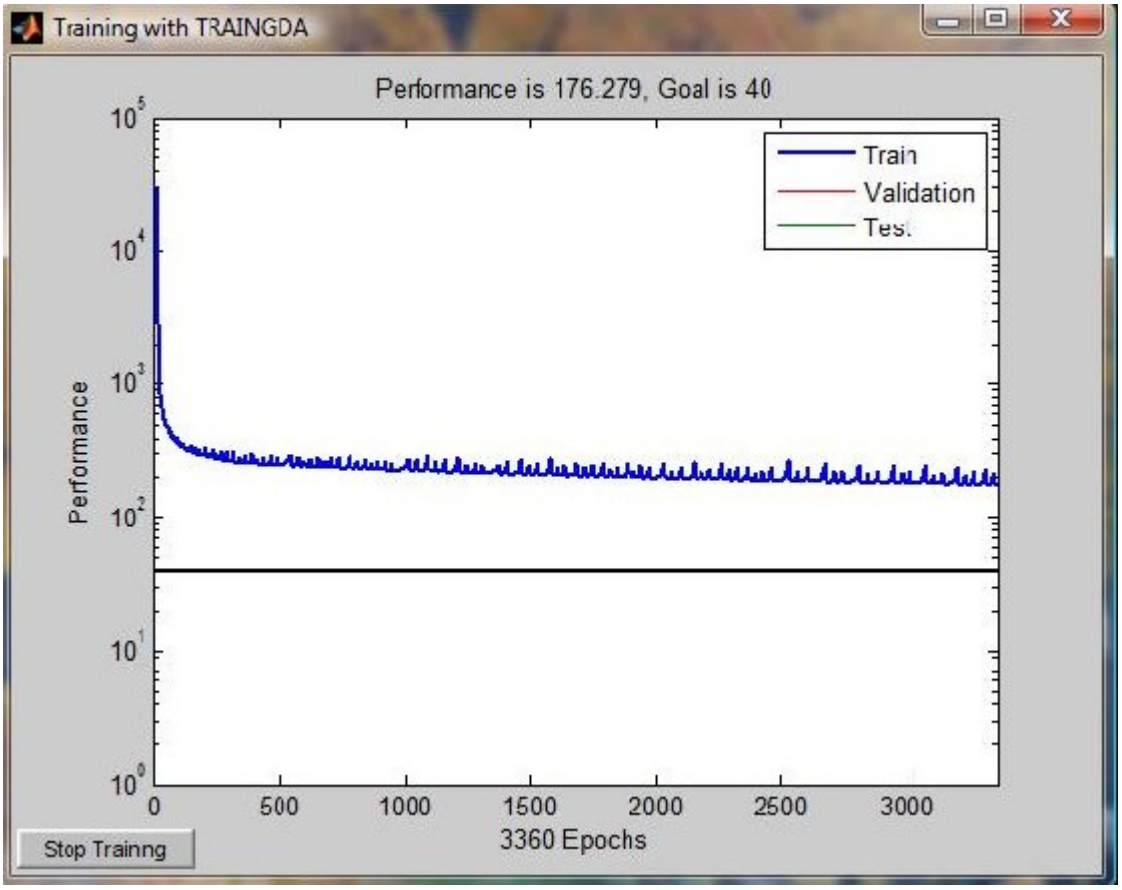


Figure 6.7 Output of training dataset using Traingda Training Function

The graph shown in figure 6.1 represents the output of the training of the network and 3360 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 176.279 .

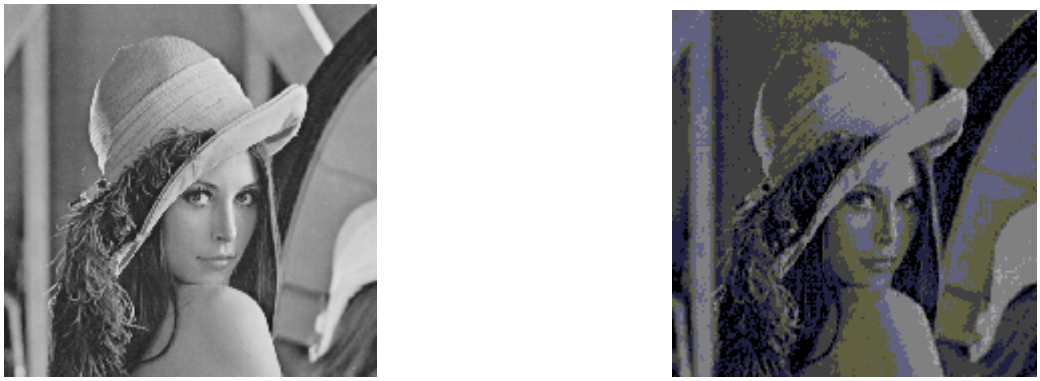


Figure 6.8 Original and Decompressed Image of Lena after 3360 epochs

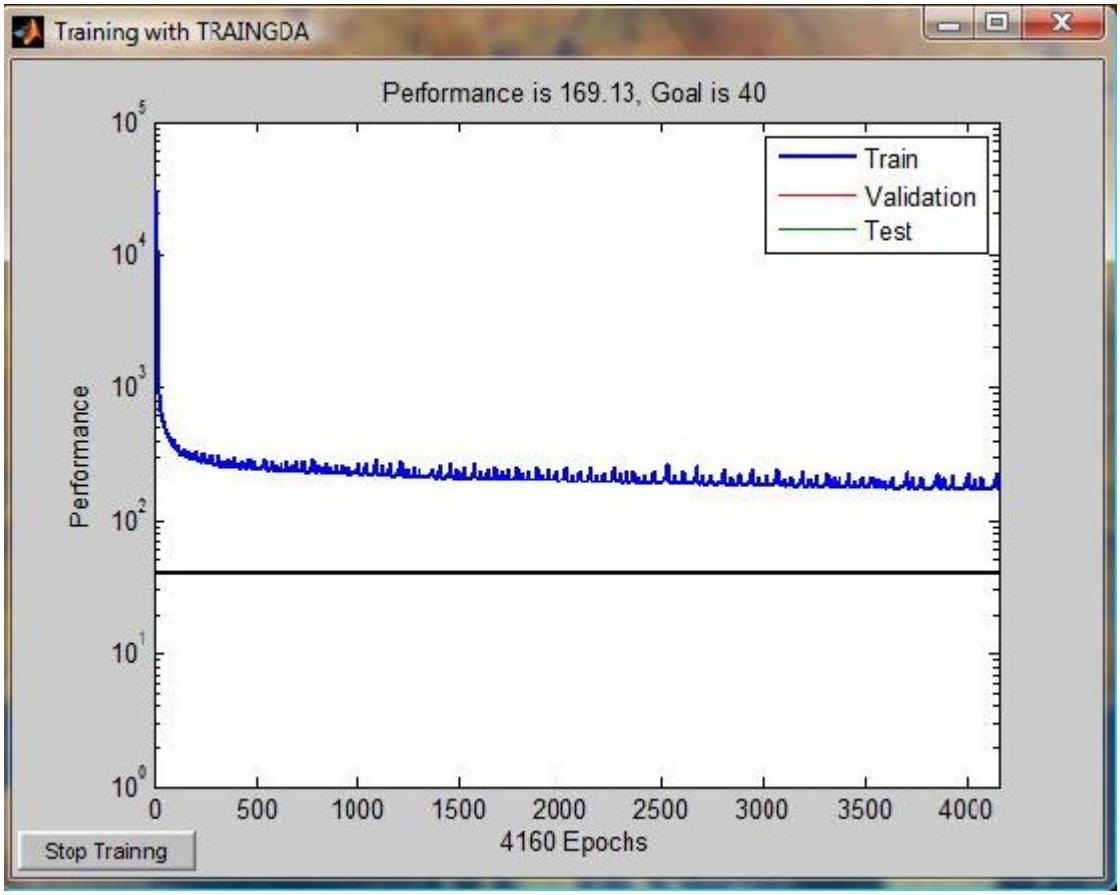


Figure 6.9 Output of training dataset using Traingda Training Function

The graph shown in figure 6.9 represents the output of the training of the network and 4160 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 169.13.

Figure 6.10 Original and Decompressed Image of Lena after 4160 epochs

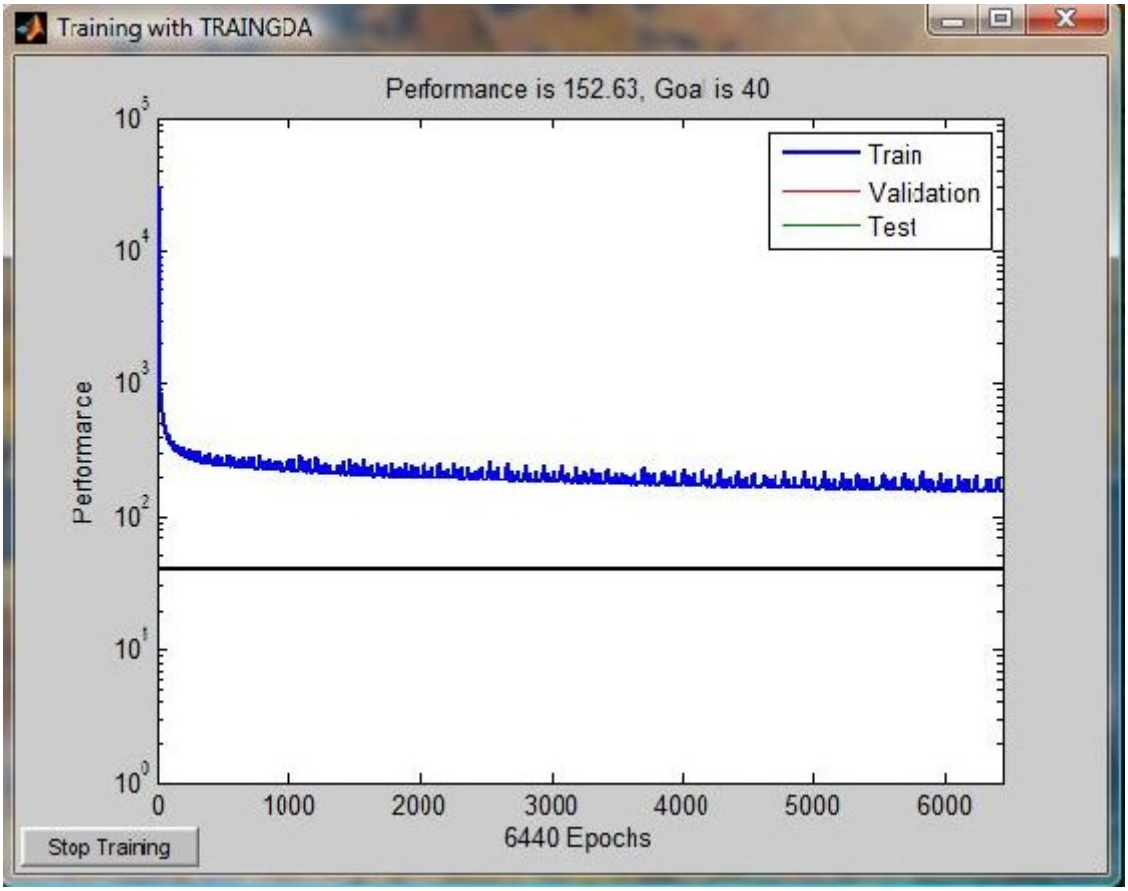


Figure 6.11 Output of training dataset using Traingda Training Function

The graph shown in figure 6.11 represents the output of the training of the network and 6440 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 152.63

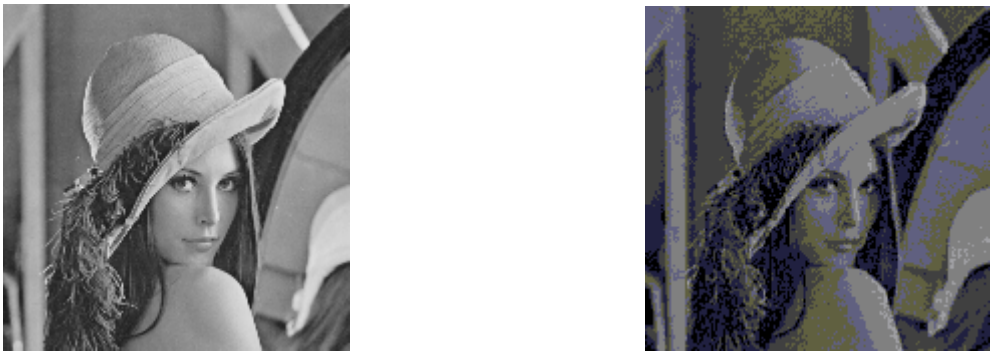


Figure 6.12 Original and Decompressed Image of Lena after 6440 epochs

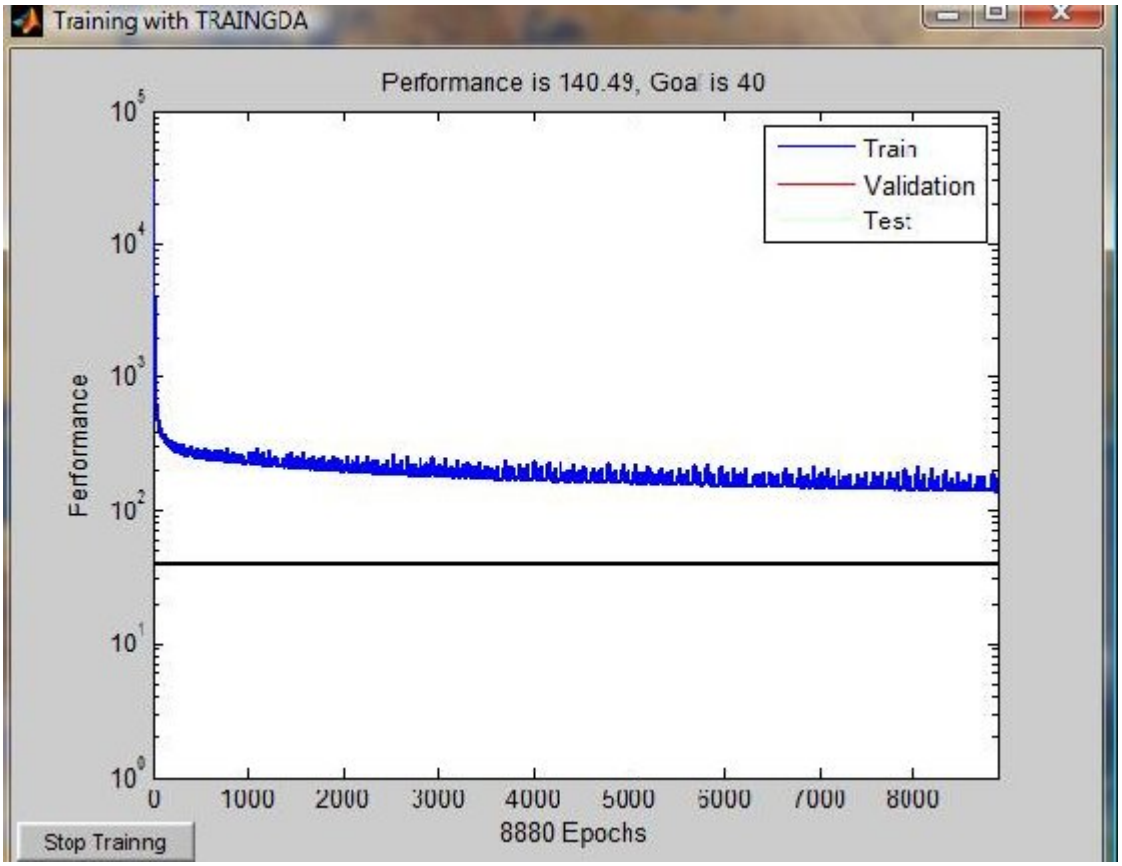


Figure 6.13 Output of training dataset using Traingda Training Function

The graph shown in figure 6.13 represents the output of the training of the network and 8880 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 140.9.

Figure 6.14 Original and Decompressed Image of Lena after 8880 epochs

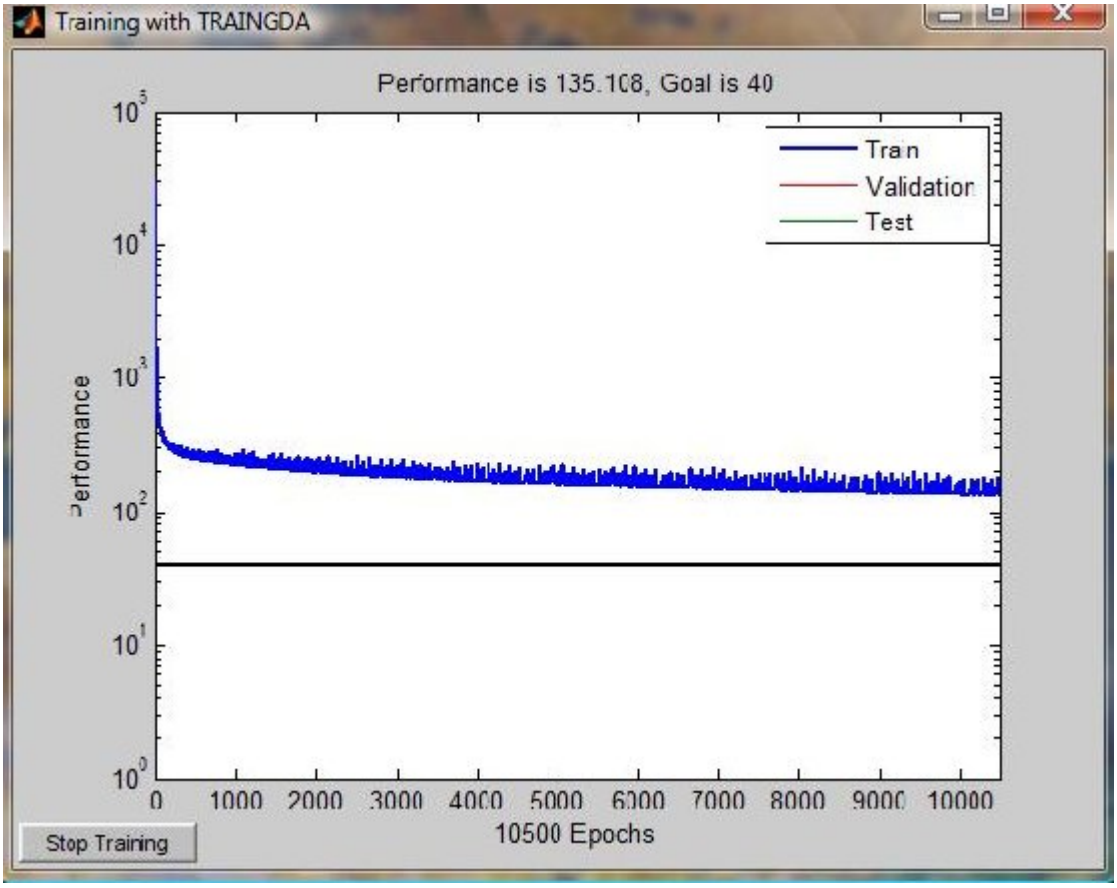


Figure 6.15 Output of training dataset using Traingda Training Function

The graph shown in figure 6.15 represents the output of the training of the network and 10500 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 135.108.

Figure 6.16 Original and Decompressed Image of Lena after 10500 epochs

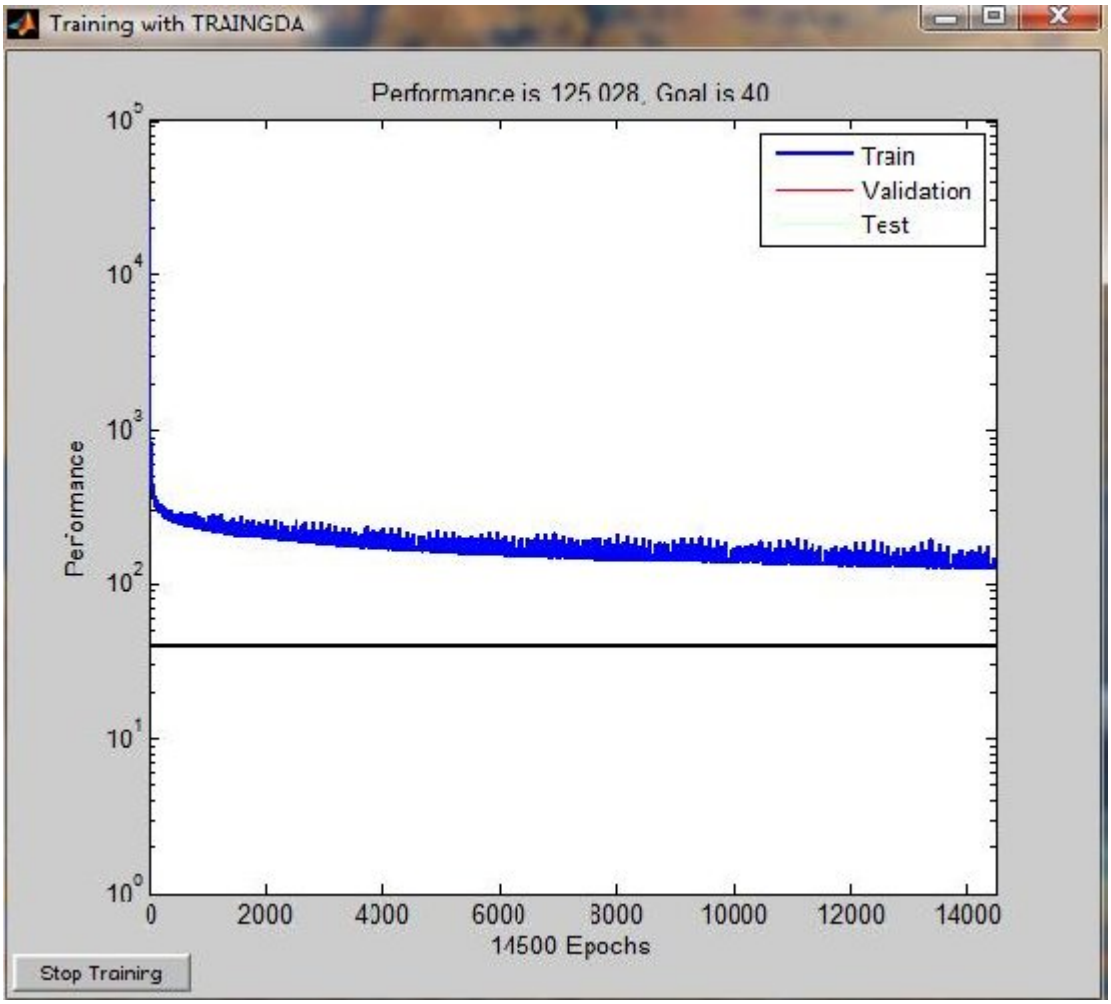


Figure 6.17 Output of training dataset using Traingda Training Function

The graph shown in figure 6.17 represents the output of the training of the network and 14500 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been125.08

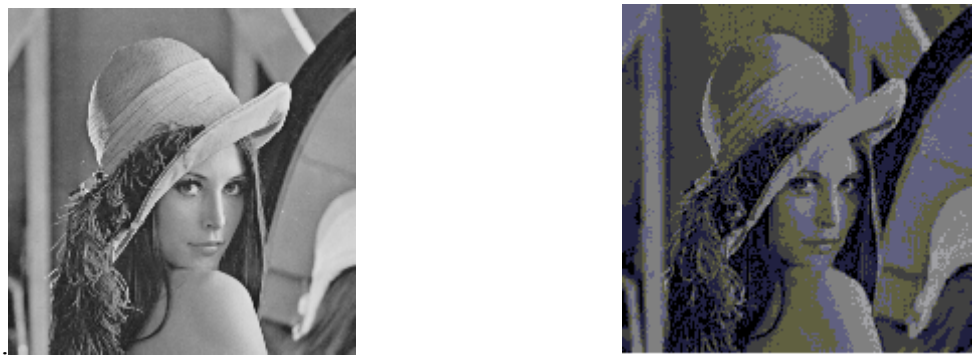


Figure 6.18 Original and Decompressed Image of Lena after 14500 epochs

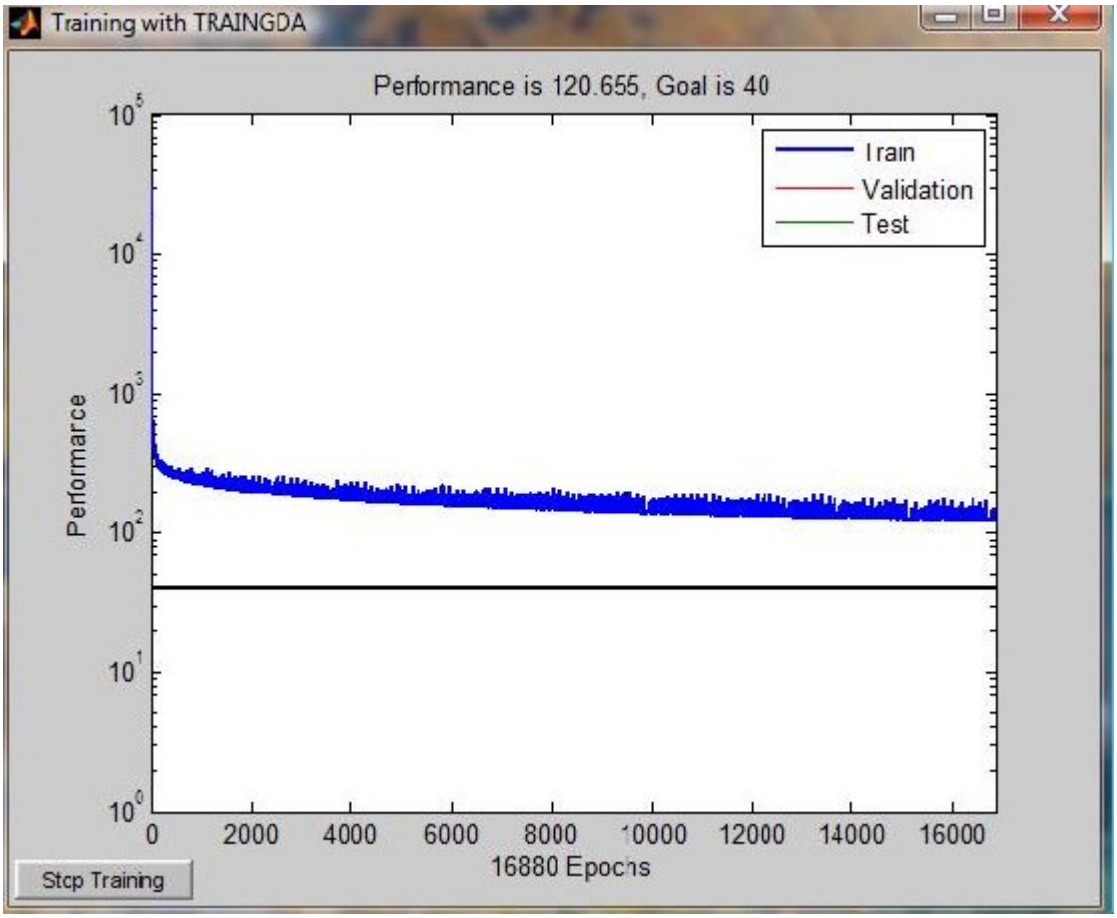


Figure 6.19 Output of training dataset using Traingda Training Function

The graph shown in figure 6.19 represents the output of the training of the network and 16880 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 120.65.

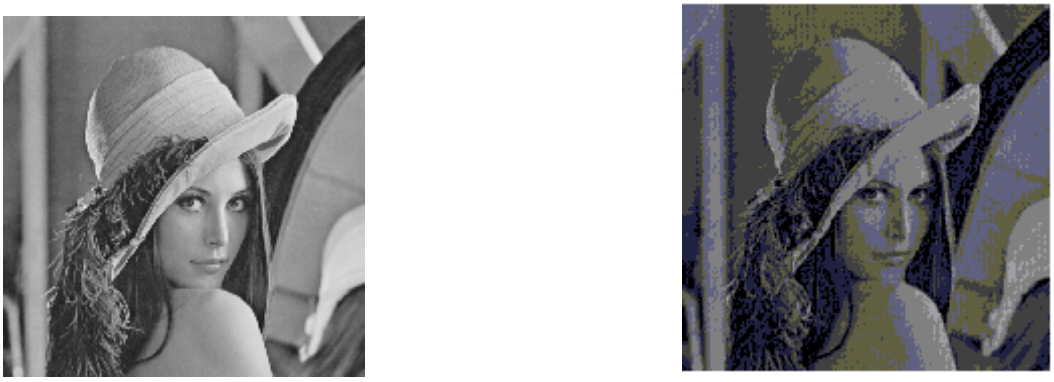


Figure 6.20 Original and Decompressed Image of Lena after 16880 epochs

Table 6.1 Different values of CR, RMSE, PSNR, B/P taken at different epochs

EPOCHSS	CR	RMSE	PSNR	B/P
860	.9972	27.9583	20.5966	.6000
1600	.9910	27.8195	20.8926	.6950
2040	.9859	27.7590	20.8992	.7179
3360	.9803	27.5021	20.9937	.7039
4160	.9762	26.8620	21.0762	.7109
6440	.9711	26.7794	21.7163	.7273
8880	.9645	26.7429	21.9732	.7401
10500	.9591	26.6784	22.0337	.7476
14500	.9521	26.3824	22.3871	.7444
16880	.9461	26.1235	22.1725	.7423

The above values of CR, RMSE, PSNR shows that images are compressed with very low loss of image quality. As the values of epochs are increasing from 860 to 16880, Compression ratio has been decreased from .9972 to .9461 and peak signal to noise ratio has been significantly increased from 20.5966 to 22.1725. This is because of network is getting more time to adjust their weight and more optimized weight are obtained to train the network.

6.3 Testing of image 2

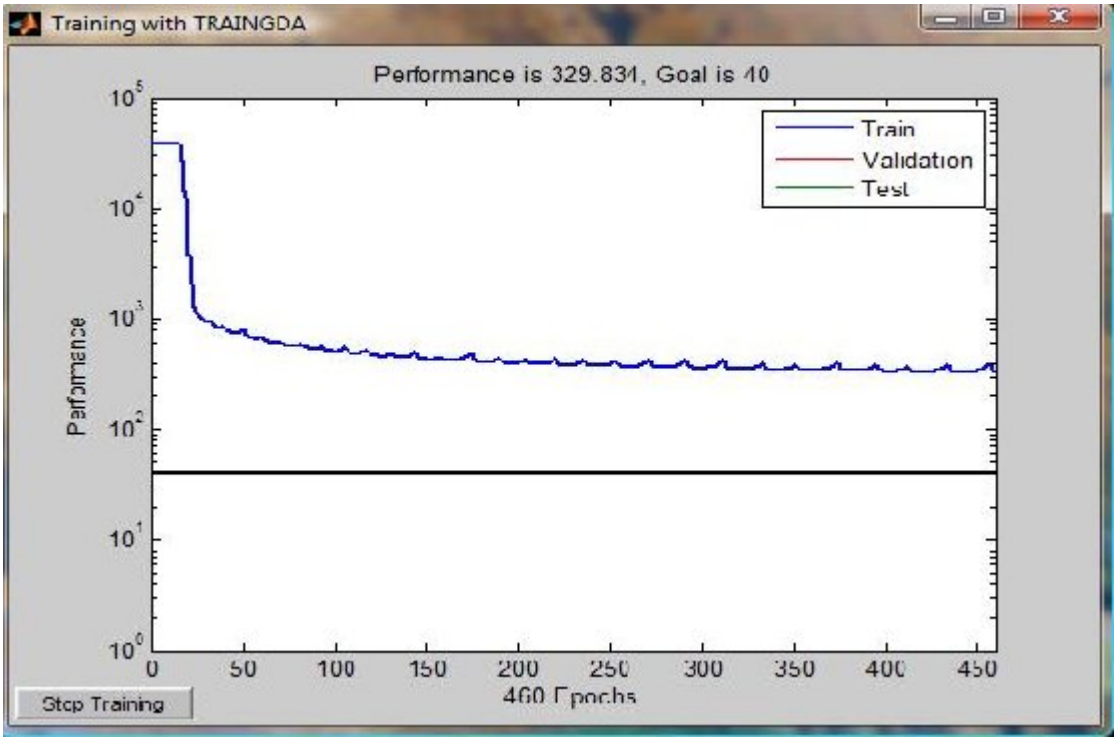


Figure 6.21 Output of training dataset using Traingda Training Function

The graph shown in figure 6.21 represents the output of the training of the network and 460 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 329.834.

Figure 6.22 Original and Decompressed Image of girl after 460 epochs

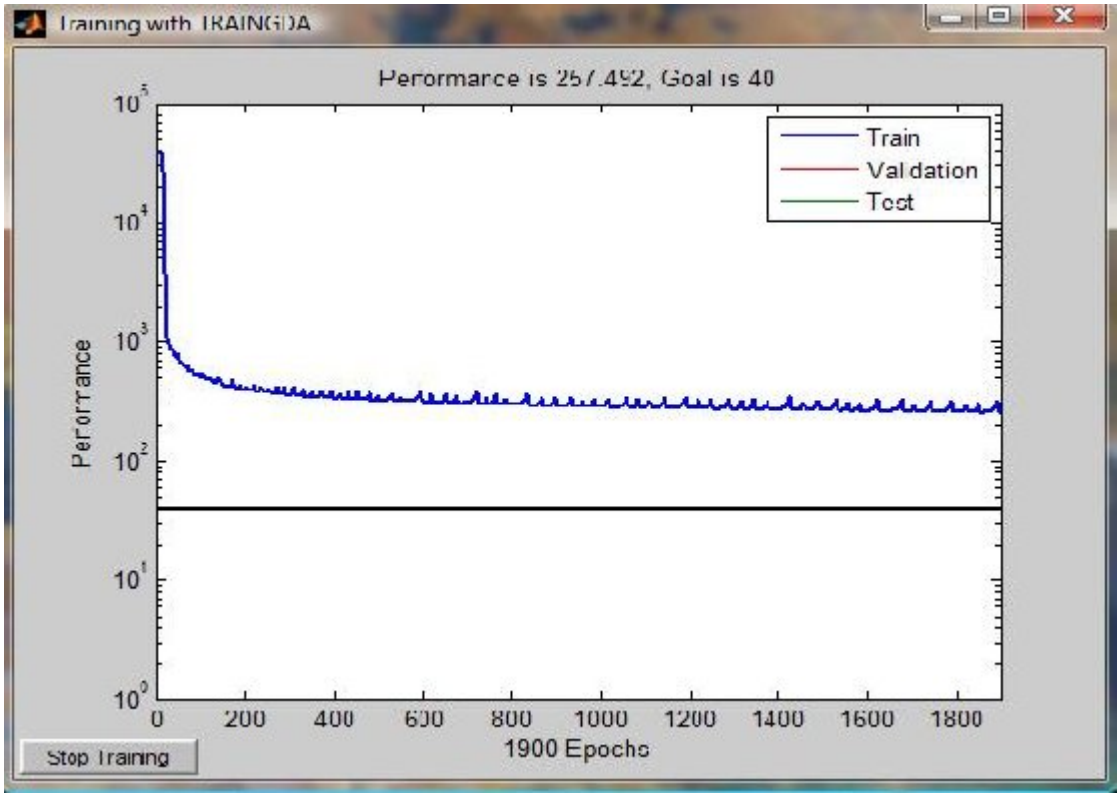


Figure 6.23 Output of training dataset using Traingda Training Function

The graph shown in figure 6.23 represents the output of the training of the network and 1900 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 257. 492.

Figure 6.24 Original and Decompressed Image Of girl after 1900 epochs

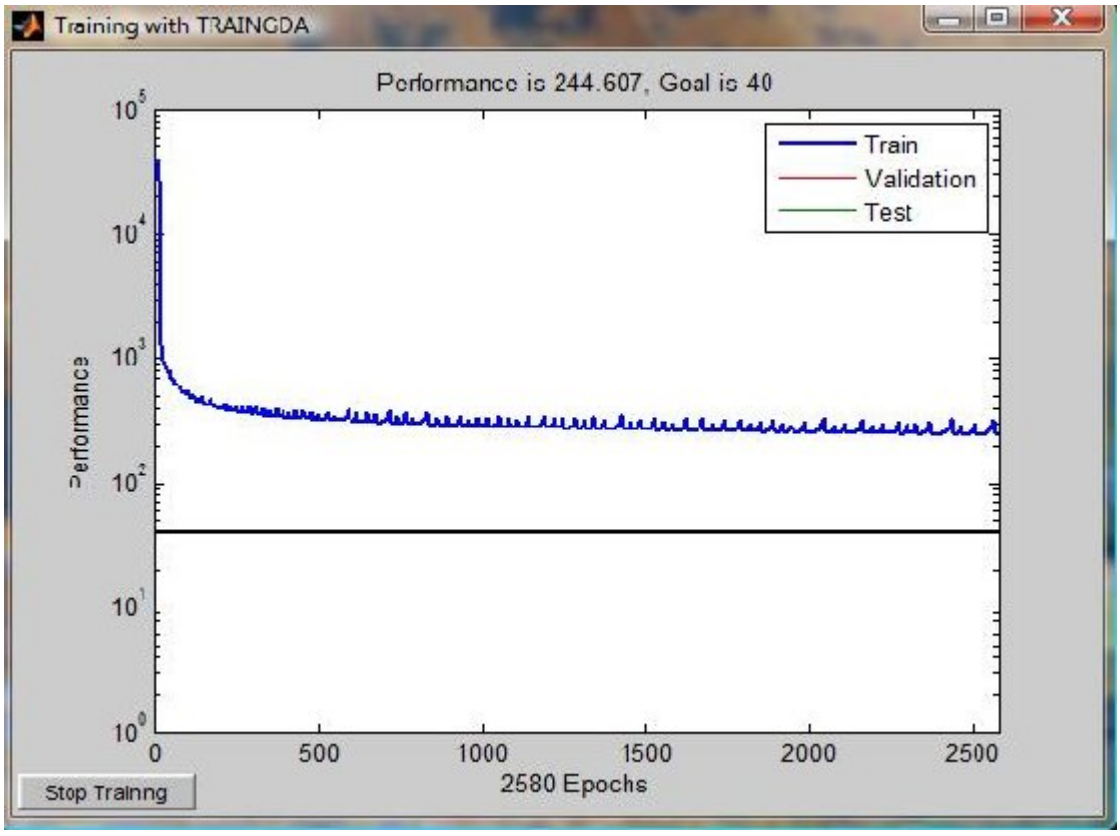


Figure 6.25 Output of training dataset using Traingda Training Function

The graph shown in figure 6.25 represents the output of the training of the network and 2580 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 244.607.

Figure 6.26 Original and Decompressed Image of girl after 2580 epochs

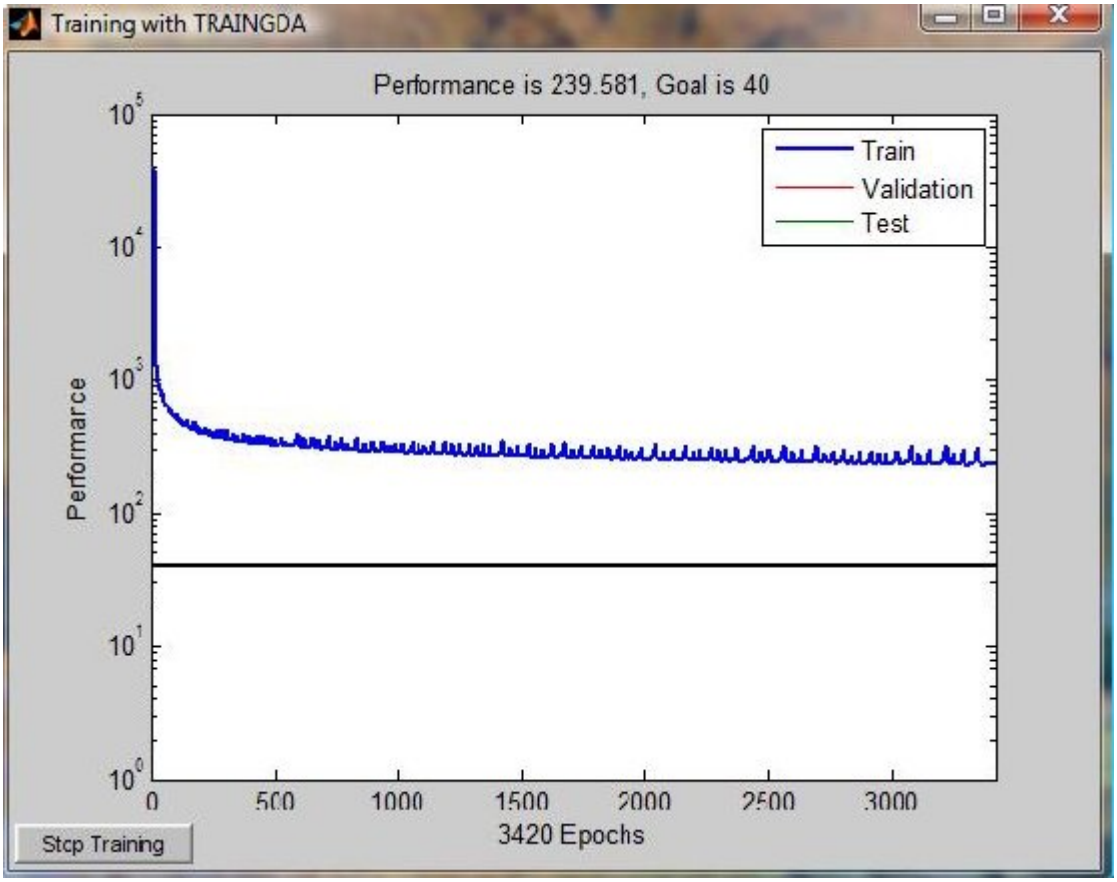


Figure 6.27 Output of training dataset using Traingda Training Function

The graph shown in figure 6.27 represents the output of the training of the network and 3420 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 239.581.

Figure 6.28 Original and Decompressed Image of girl after 3420

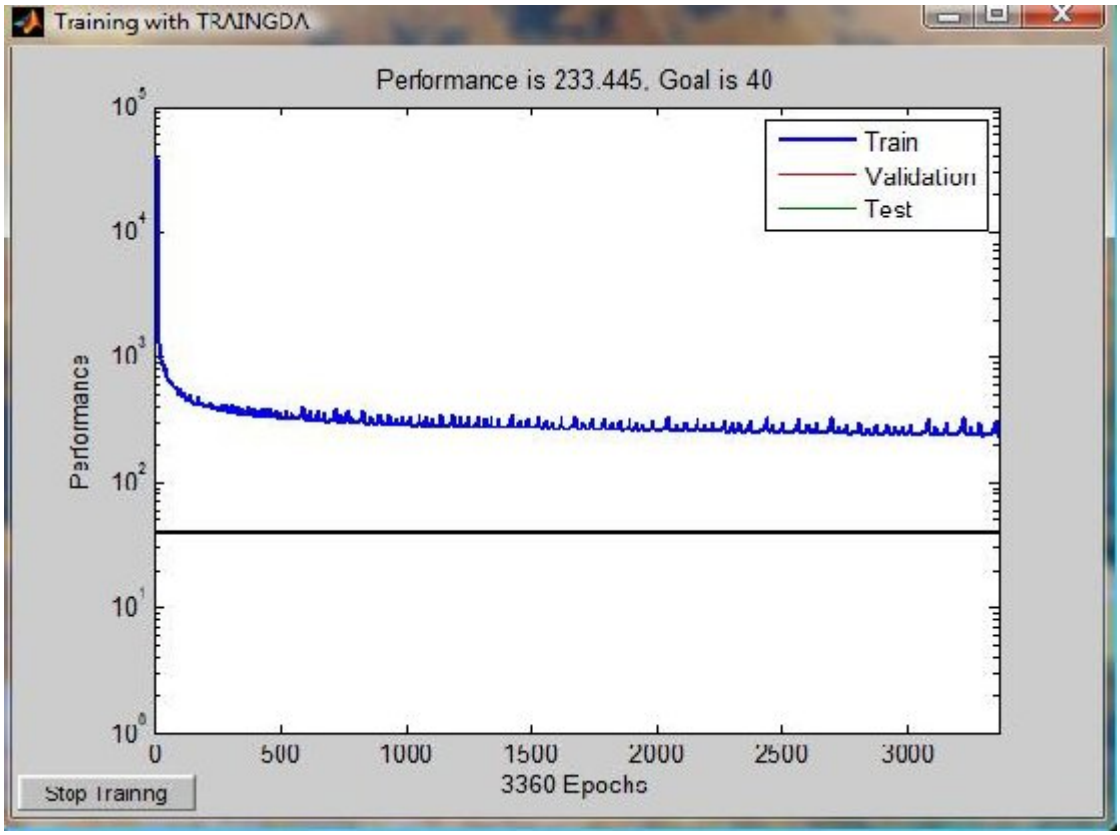


Figure 6.29 Output of training dataset using Traingda Training Function

The graph shown in figure 6.29 represents the output of the training of the network and 3360 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 233.445.

Figure 6.30 Original and Decompressed Image of girl after 3360 epochs

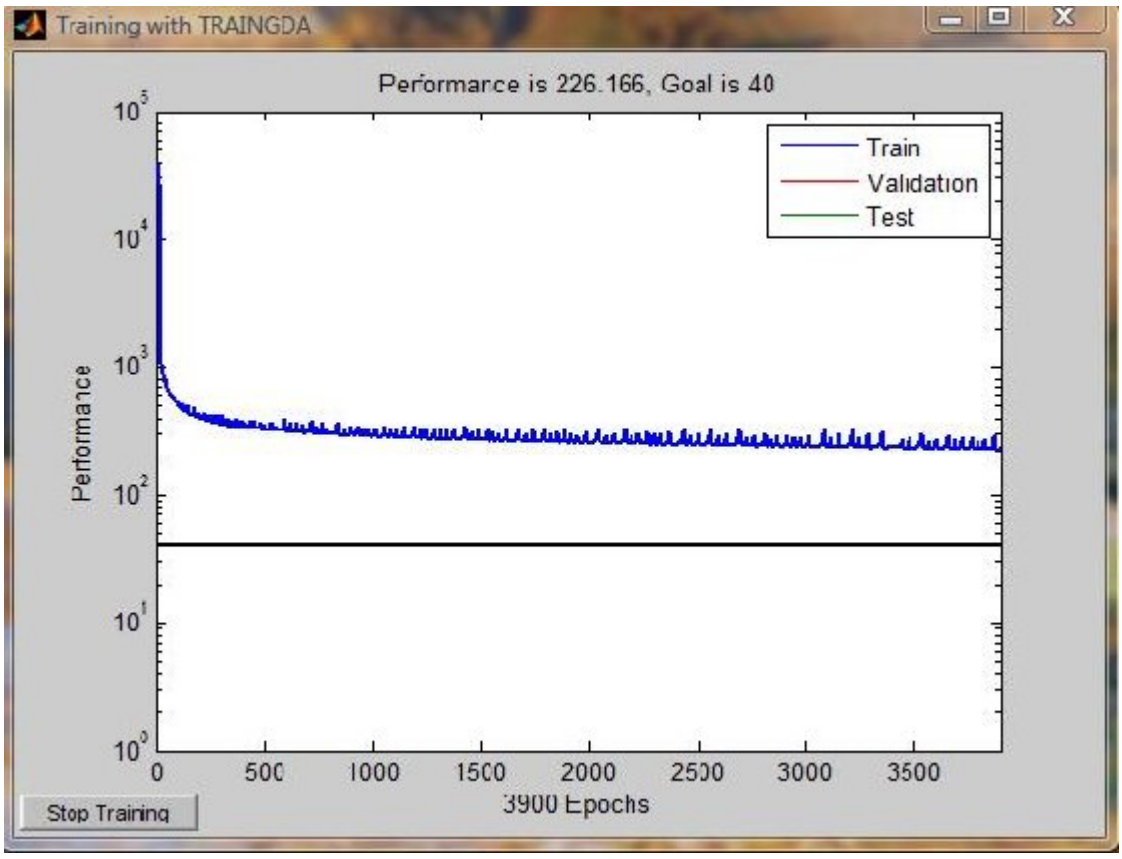


Figure 6.31 Output of training dataset using Traingda Training Function

The graph shown in figure 6.31 represents the output of the training of the network and 3900 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 226.166 .

Figure 6.32 Original and Decompressed Image of girl after 3900 epochs

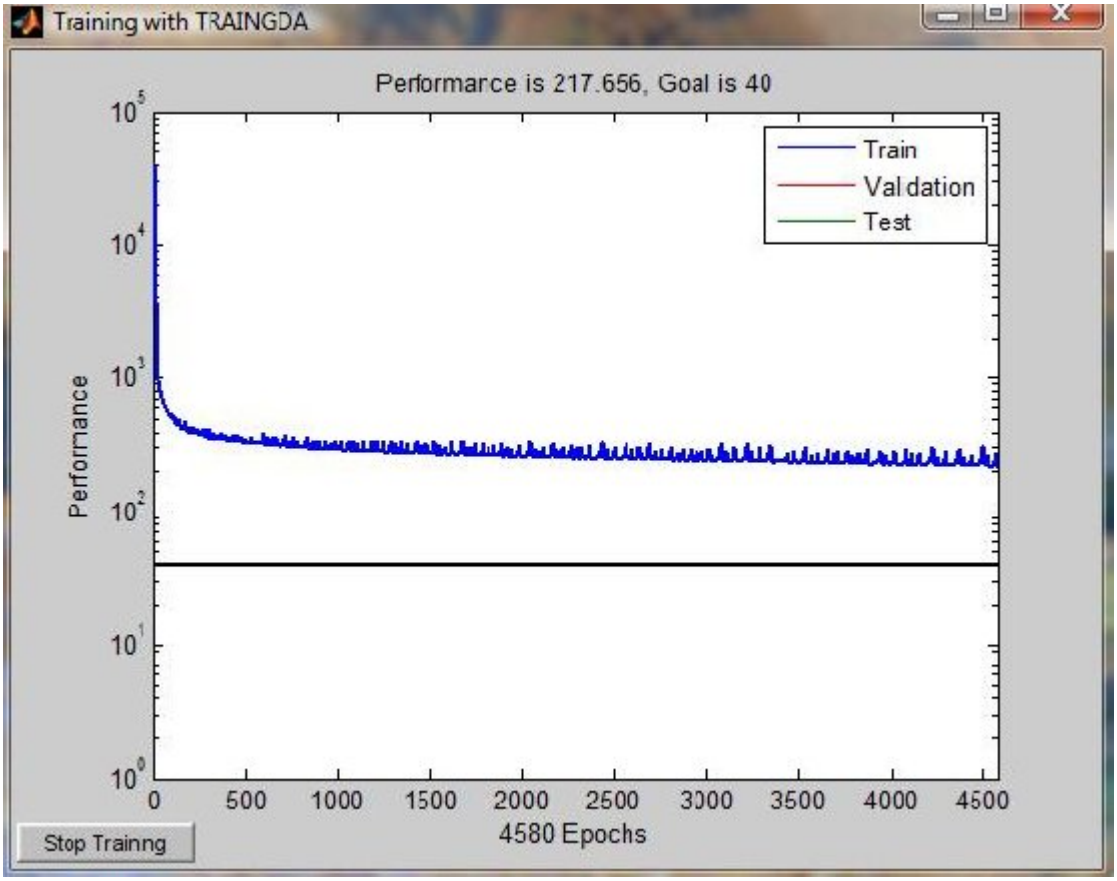


Figure 6.33 Output of training dataset using Traingda Training Function

The graph shown in figure 6.33 represents the output of the training of the network and 4580 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 217.665.

Figure 6.34 Original and Decompressed Image of girl after 4580 epochs

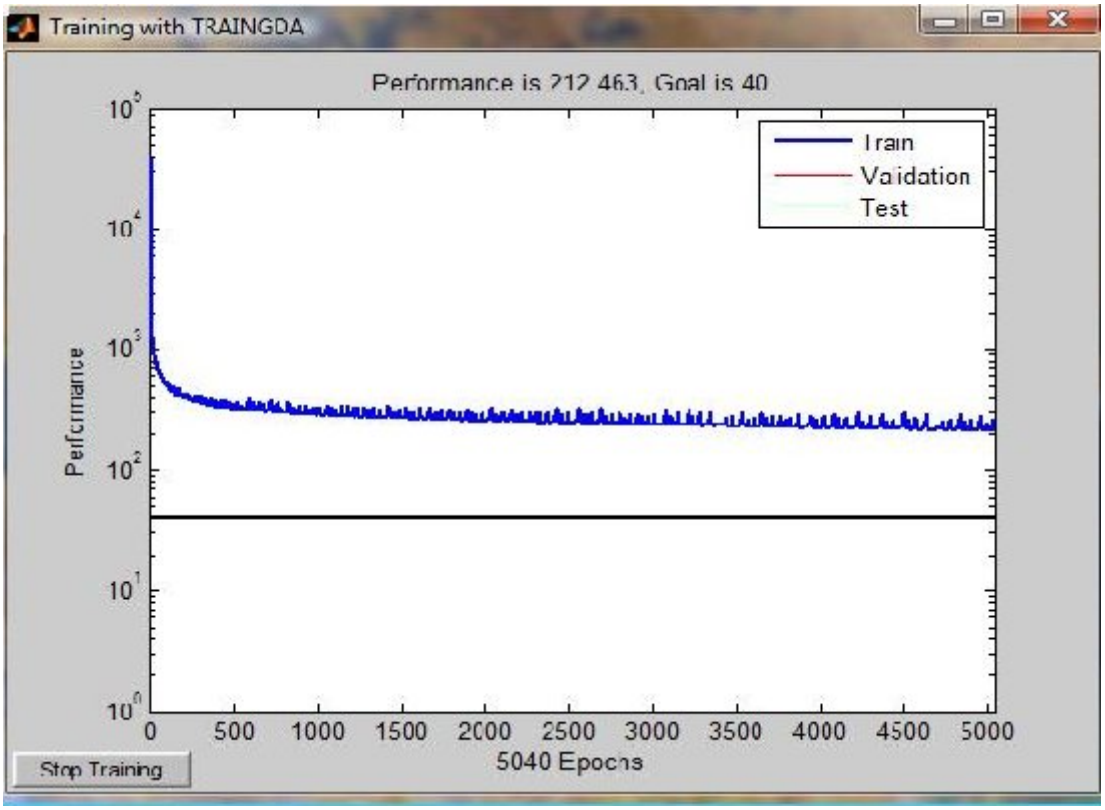


Figure 6.35 Output of training dataset using Traingda Training Function

The graph shown in figure 6.35 represents the output of the training of the network and 5040 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 212.463 .

Figure 6.36 Original and Decompressed Image of girl after 5040 epochs

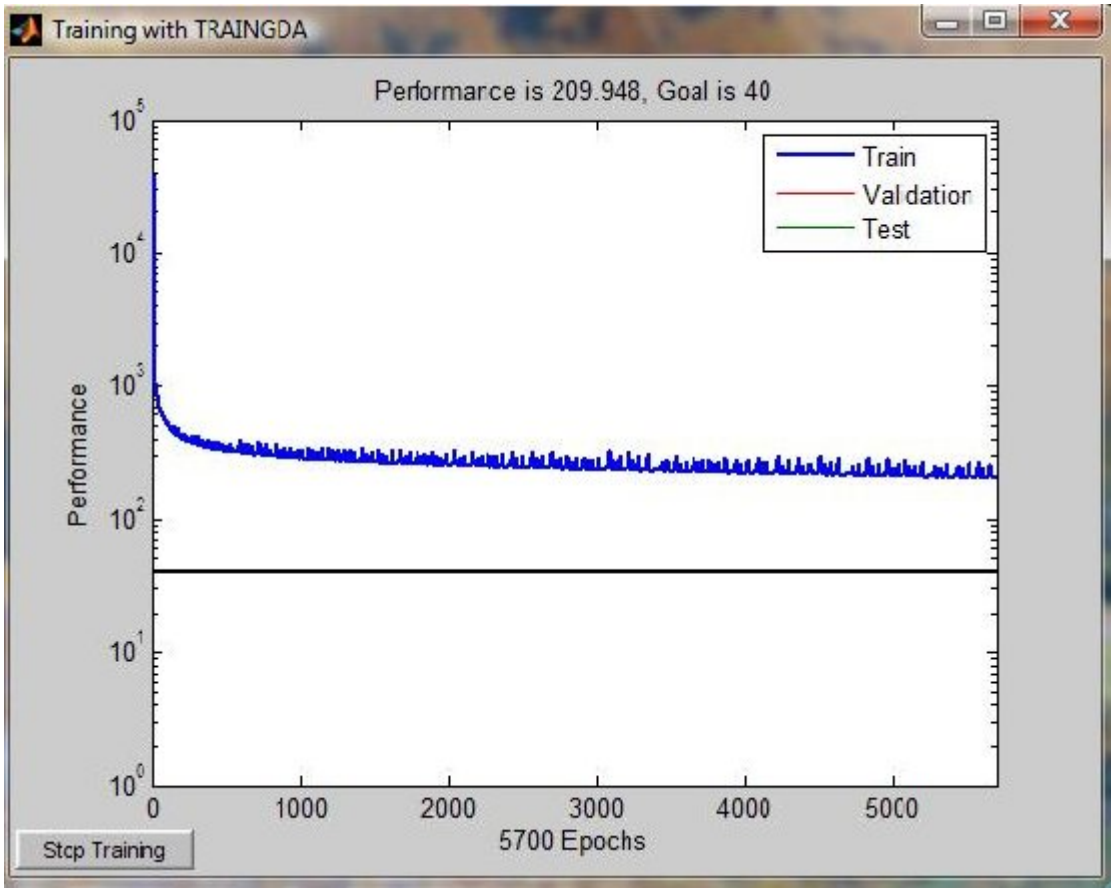


Figure 6.37 Output of training dataset using Traingda Training Function

The graph shown in figure 6.37 represents the output of the training of the network and 5700 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 209.948

Figure 6.38 Original and Decompressed Image of girl after 5700 epochs

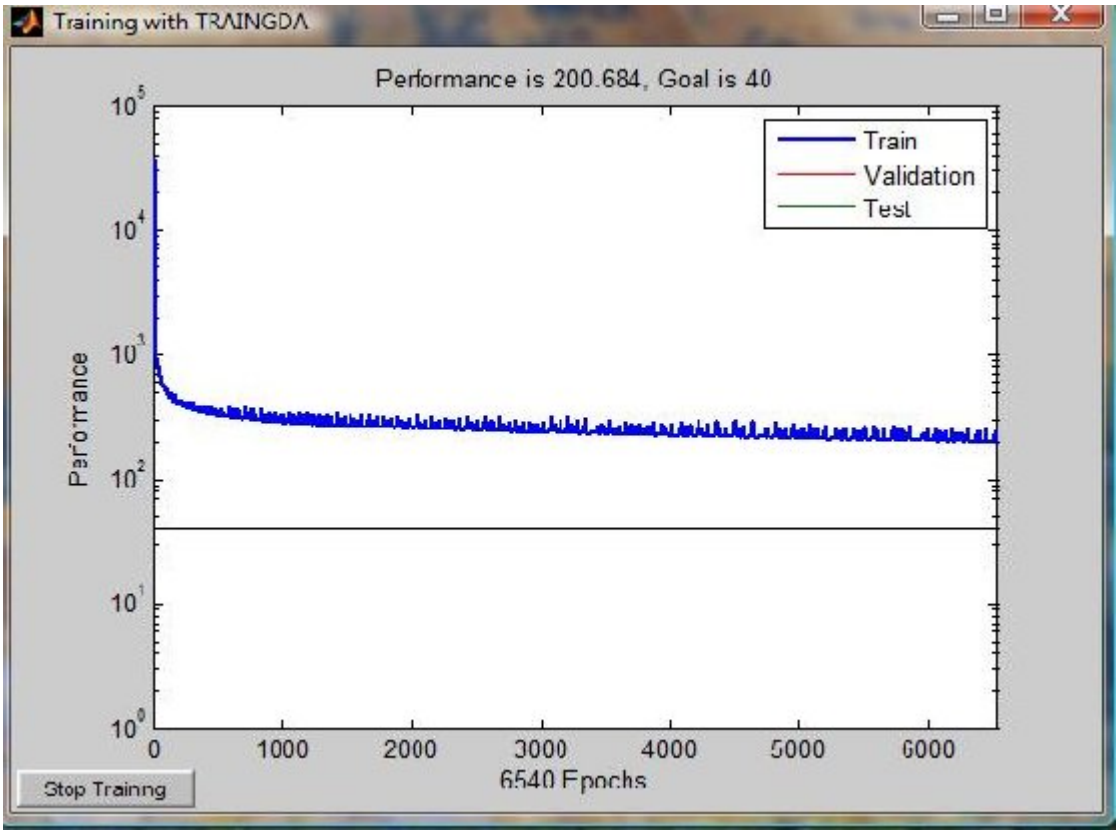


Figure 6.39 Output of training dataset using Traingda Training Function

The graph shown in figure 6.39 represents the output of the training of the network and 6540 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 200.684.

Figure 6.40 Original and Decompressed Image of girl after 6540 epochs

Table 6.2 Different values of CR, RMSE, PSNR, B/P taken at different epochs

EPOCHS	CR	RMSE	PSNR	B/P
460	.9965	27.0399	19.2283	.5779
1900	.9912	27.4554	20.3641	.5833
2580	.9887	27.3567	20.1213	.5887
3420	.9810	27.2926	19.1346	.6000
3360	.9776	26.9449	19.8734	.6043
3900	.9721	26.8445	20.4740	.6154
4580	.9645	26.6055	20.8217	.6246
5040	.9559	26.4231	20.8858	.6292
5700	.9510	26.3350	20.9845	.6350
6540	.9417	25.1356	20.5690	.6465

The above values of CR, RMSE, PSNR shows that image is compressed with very low loss of image quality. As the values of epochs are increasing from 460 to 6540, Compression ratio has been reduced from .9965 to .9417 and peak signal to noise ratio has been increased from 19.2283 to 20.5690. This is because of network is getting more time to adjust their weight and more optimized weight are obtained to train the network.

6.4 Testing of image 3

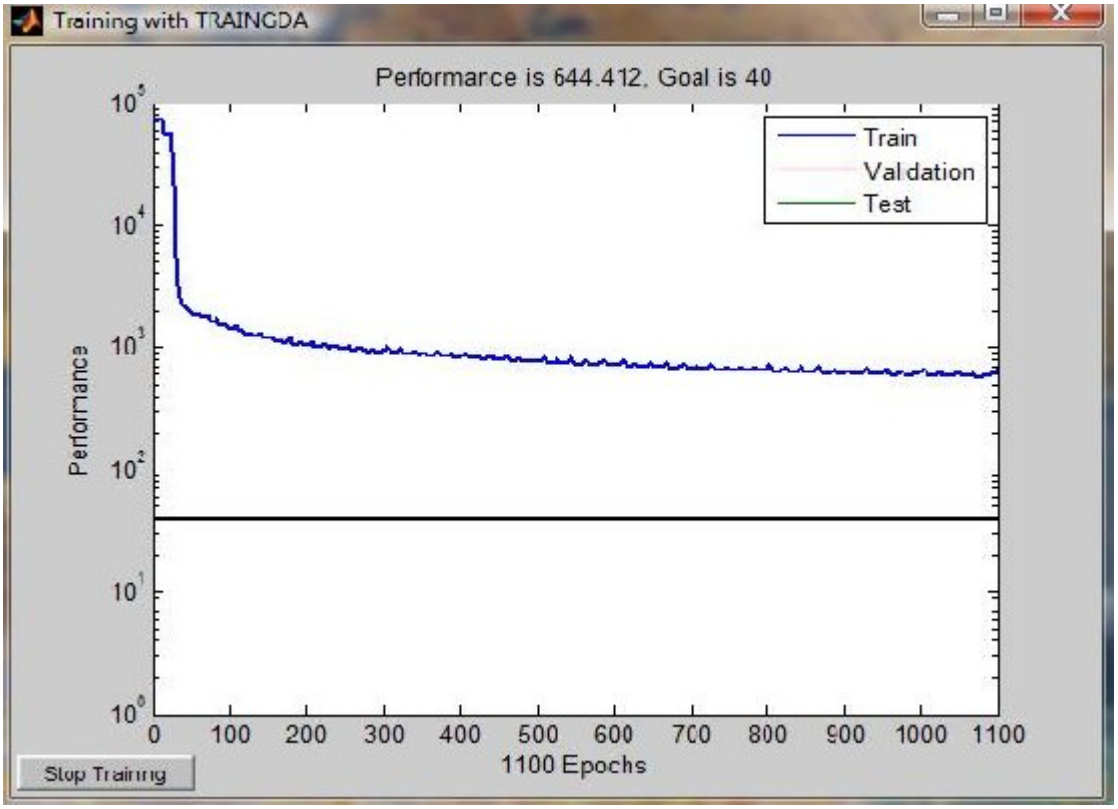


Figure 6.41 Output of training dataset using Traingda Training Function

The graph shown in figure 6.41 represents the output of the training of the network and 1100 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 644.412.

Figure 6.42 Original And Decompressed Image Of cameraman after 1100 epochs

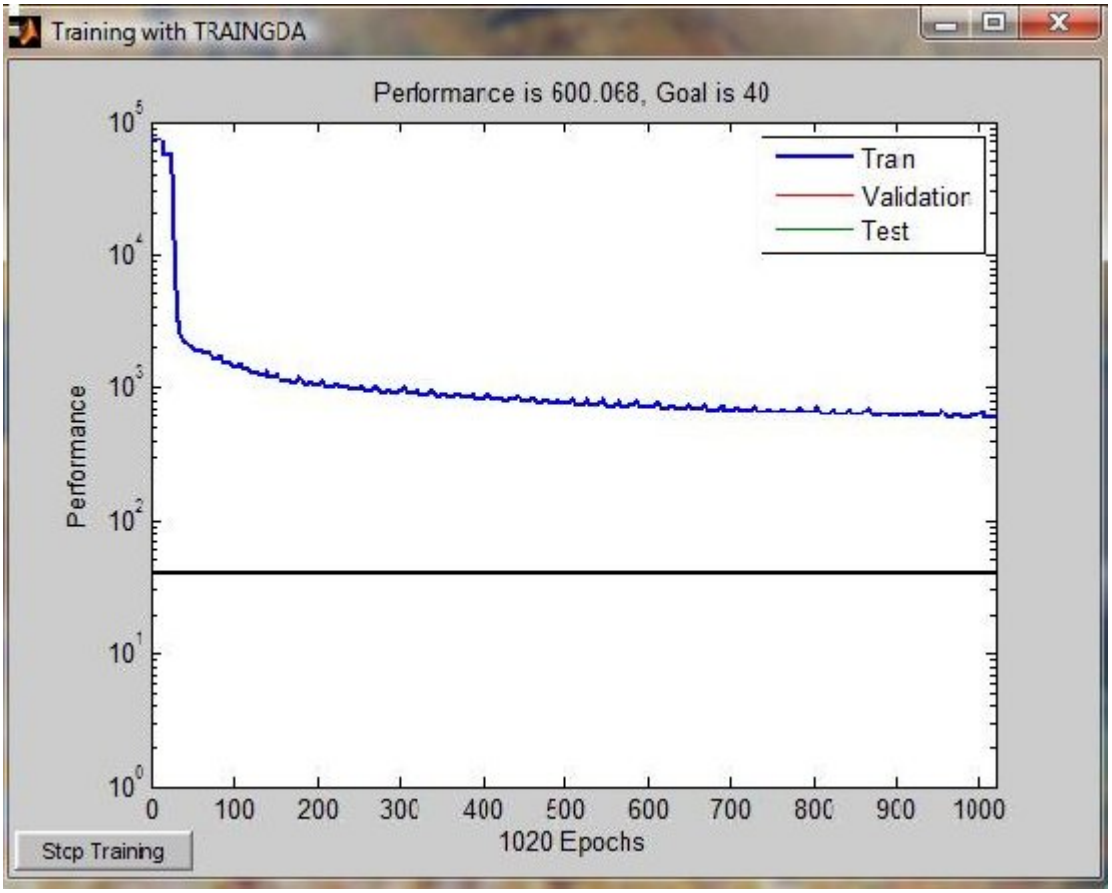


Figure 6.43 Output of training dataset using Traingda Training Function

The graph shown in figure 6.43 represents the output of the training of the network and 1020 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 600.068.

Figure 6.44 Original and Decompressed Image of cameraman after 1020 epochs

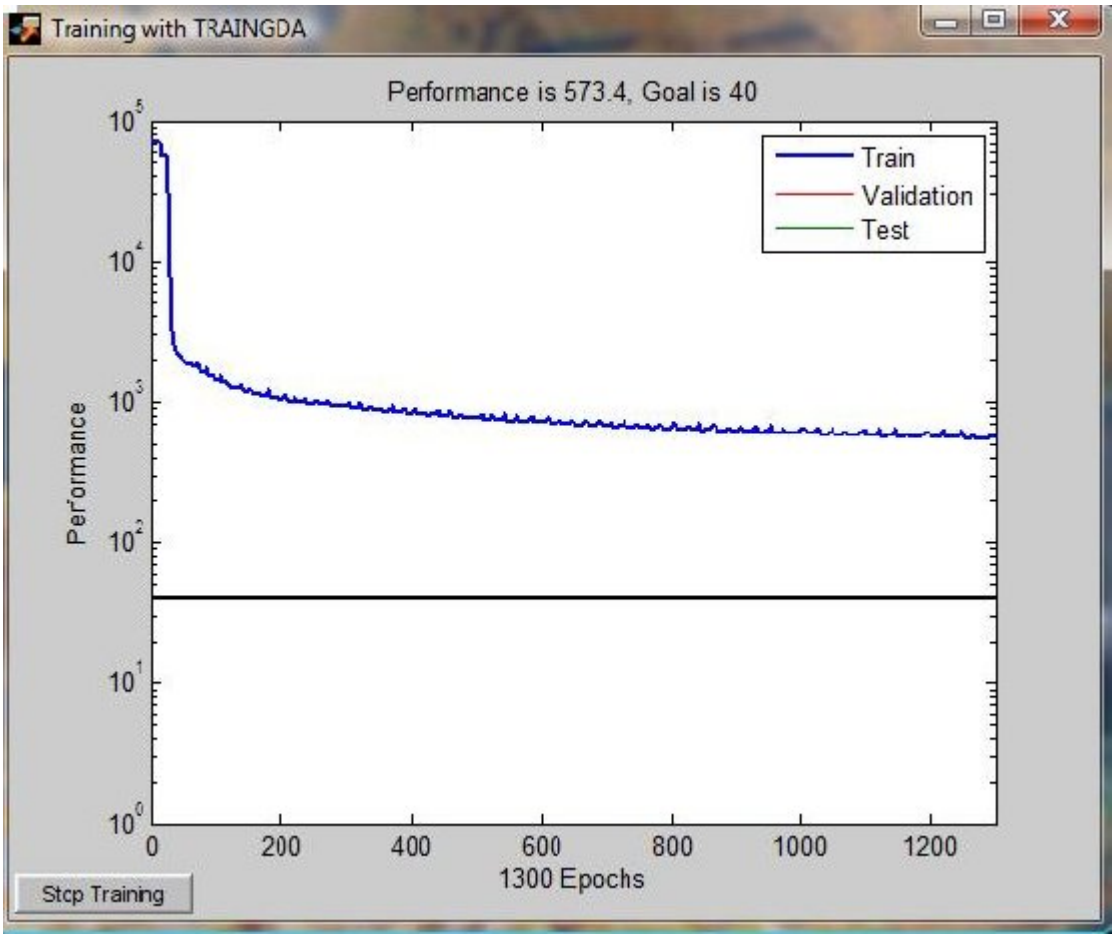


Figure 6.45 Output of training dataset using Traingda Training Function

The graph shown in figure 6.45 represents the output of the training of the network and 1300 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 573.4.

Figure 6.46 Original and Decompressed Image of cameraman after 1300 epochs

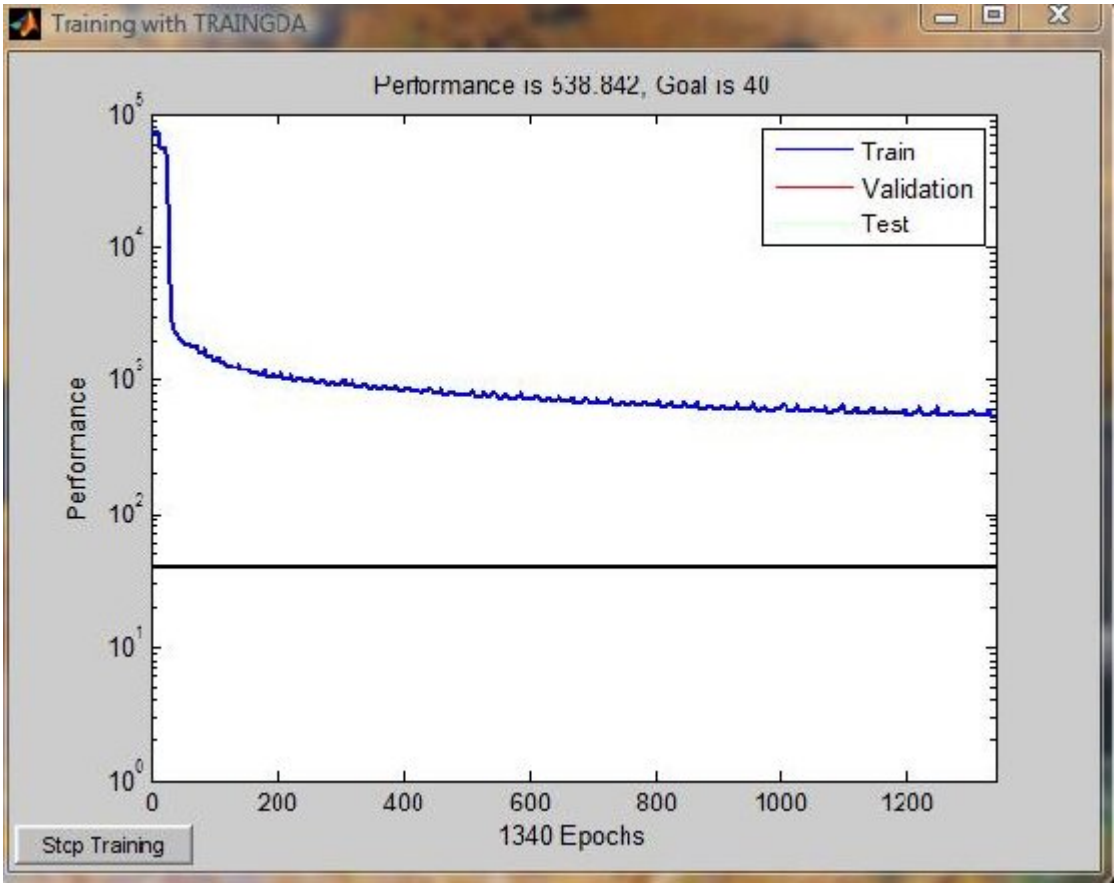


Figure 6.47 Output of training dataset using Traingda Training Function

The graph shown in figure 6.47 represents the output of the training of the network and 1340 epochs have been taken to get trained the network using the trainlm train function. In this case the performance goal of the network has been 538.842.

Figure 6.48 Original And Decompressed Image Of cameraman after 1340 epochs

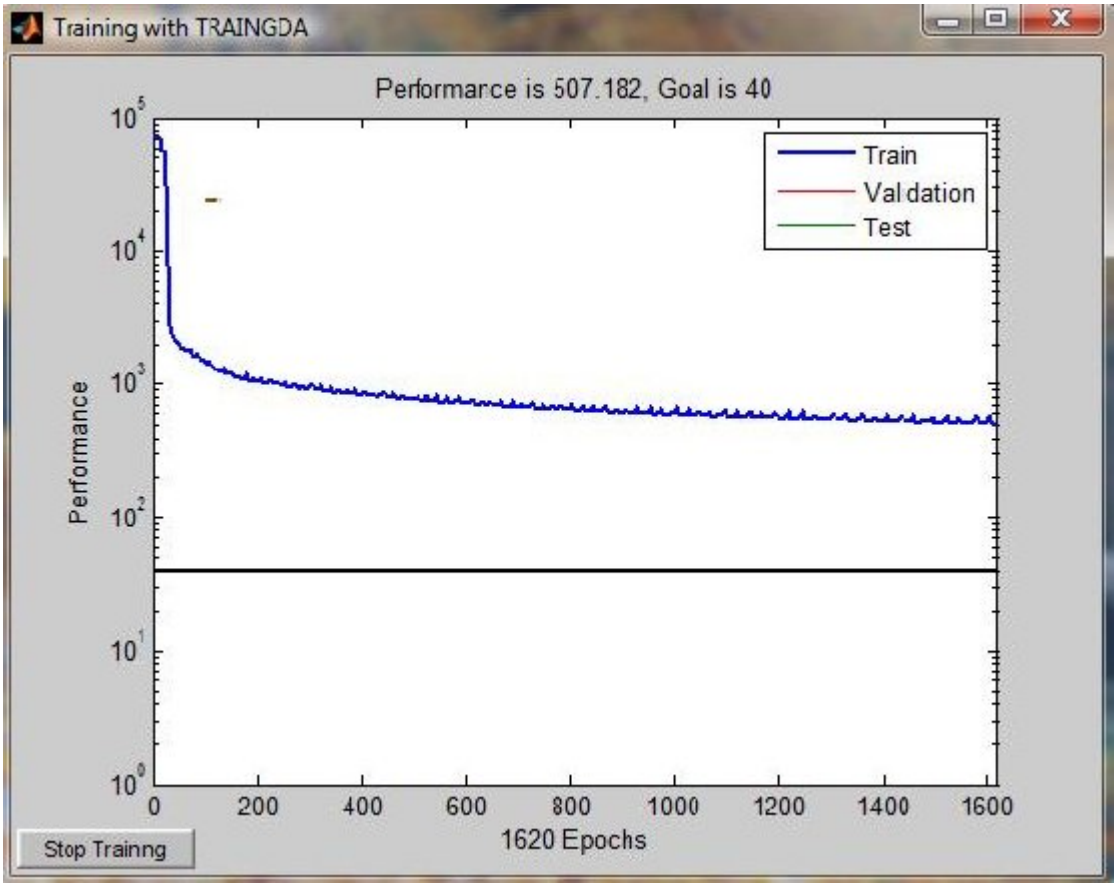


Figure 6.49 Output of training dataset using Traingda Training Function

The graph shown in figure 6.49 represents the output of the training of the network and 1620 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 507.182.

Figure 6.50 Original And Decompressed Image Of cameraman after 1620 epochs

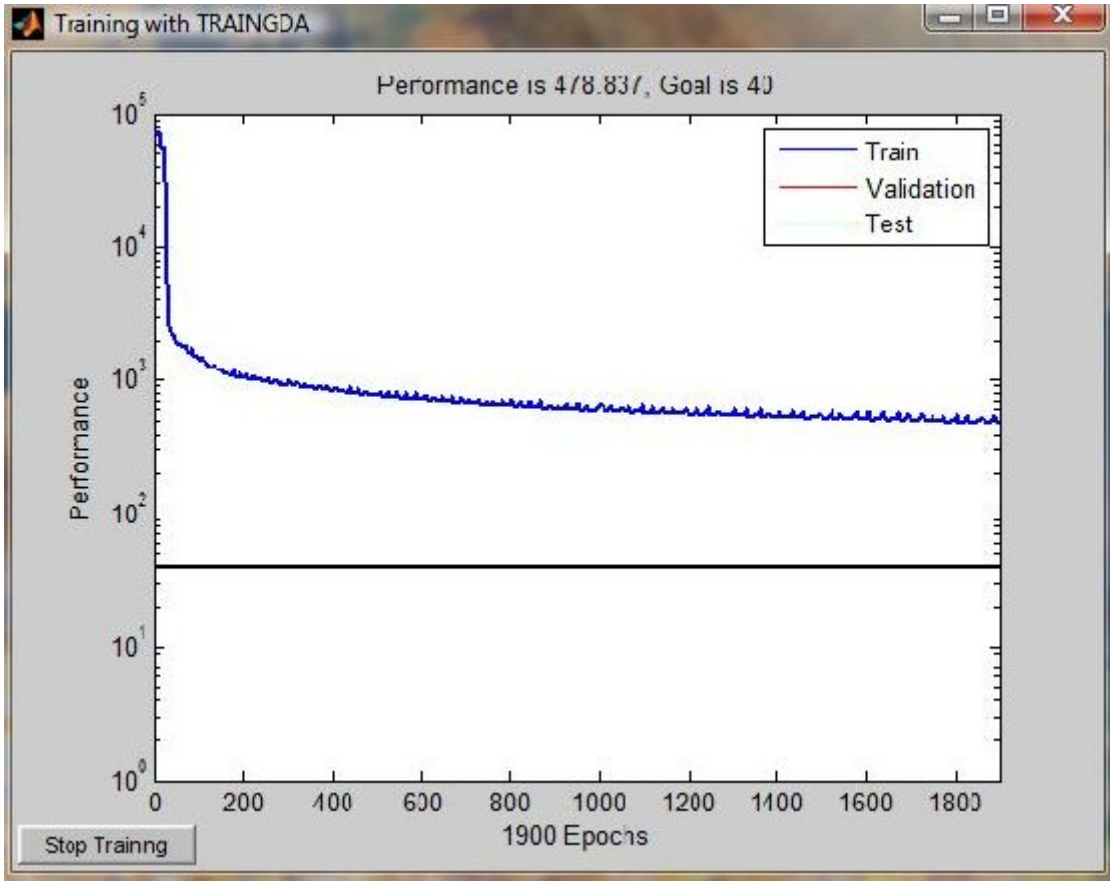


Figure 6.51 Output of training dataset using Traingda Training Function

The graph shown in figure 6.51 represents the output of the training of the network and 1900 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 478.837.

Figure 6.52 Original and Decompressed Image of cameraman after 1900 epochs

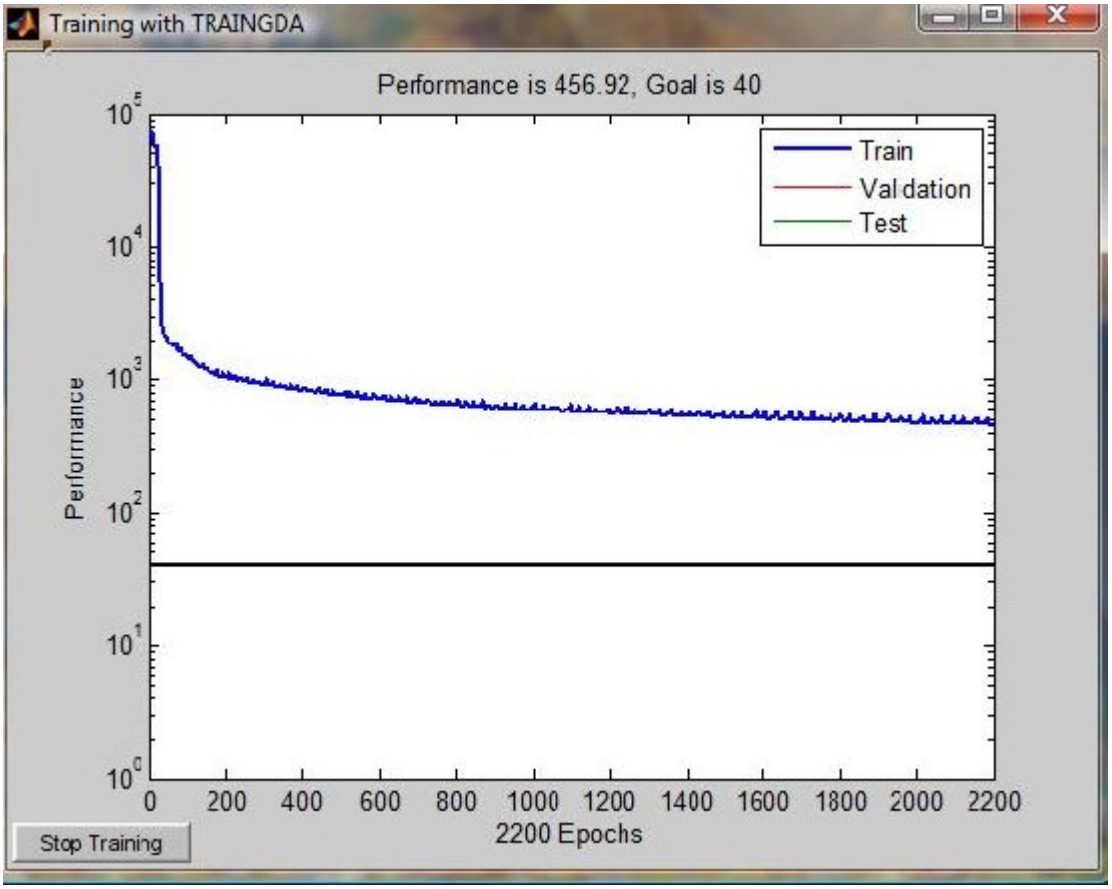


Figure 6.53 Output of training dataset using Traingda Training Function

The graph shown in figure 6.53 represents the output of the training of the network and 2200 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 456.92.

Figure 6.54 Original And Decompressed Image Of cameraman after 2200 epochs

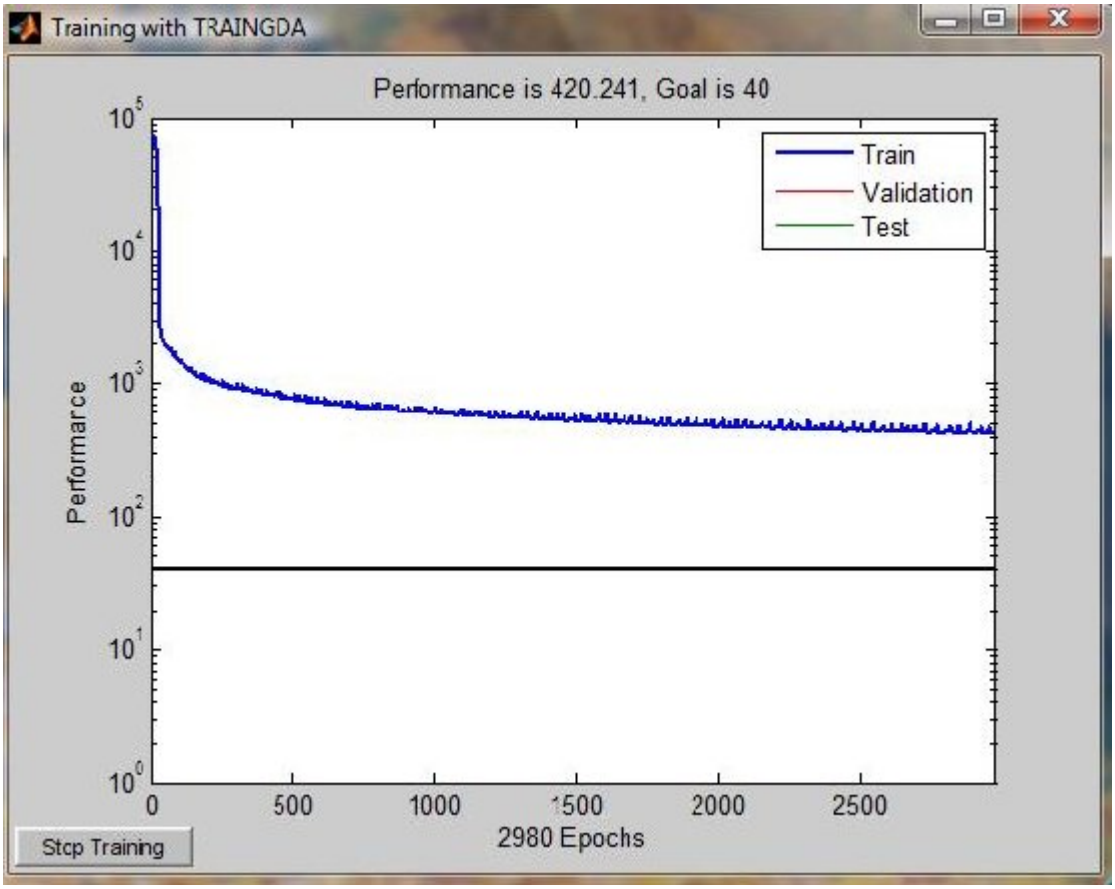


Figure 6.55 Output of training dataset using Traingda Training Function

The graph shown in figure 6.55 represents the output of the training of the network and 2980 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 420.241.

Figure 6.56 Original and Decompressed Image of cameraman after 2980 epochs

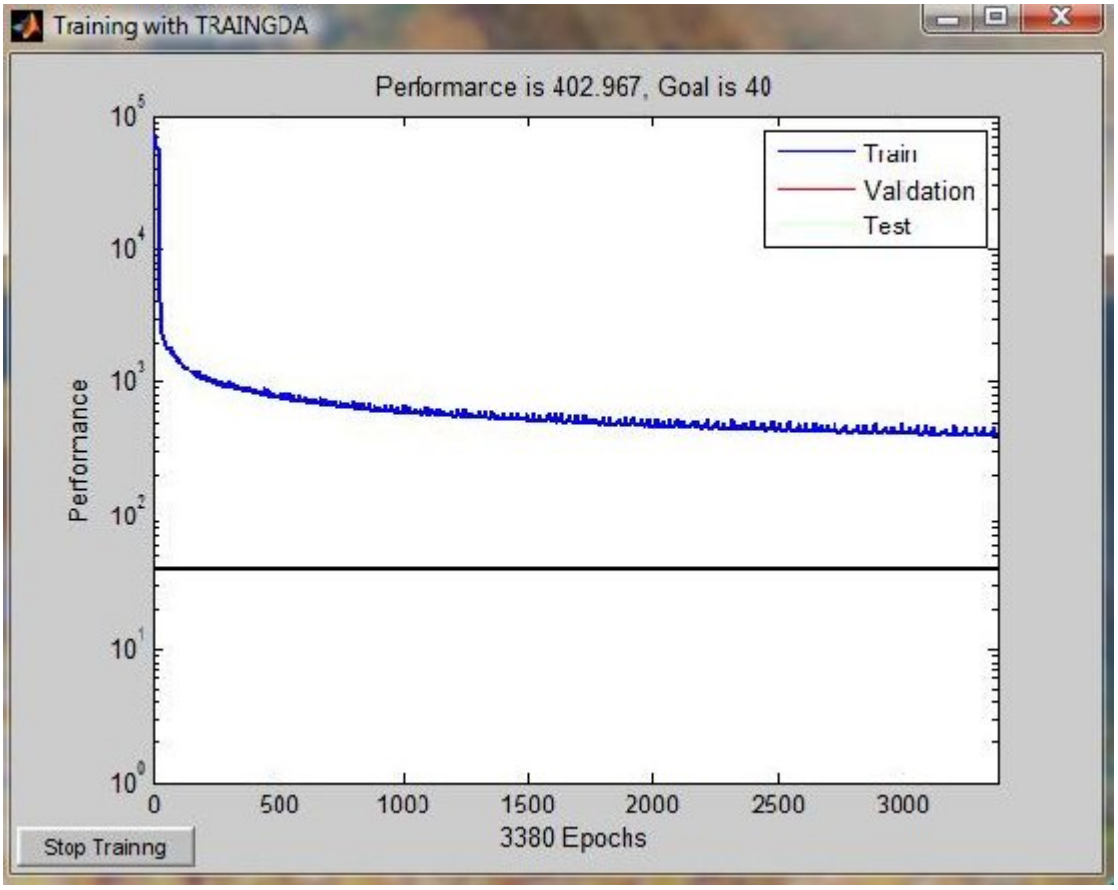


Figure 6.57 Output of training dataset using Traingda Training Function

The graph shown in figure 6.57 represents the output of the training of the network and 3380 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 402. 967.

Figure 6.58 Original asnd Decompressed Image of cameraman after 3380

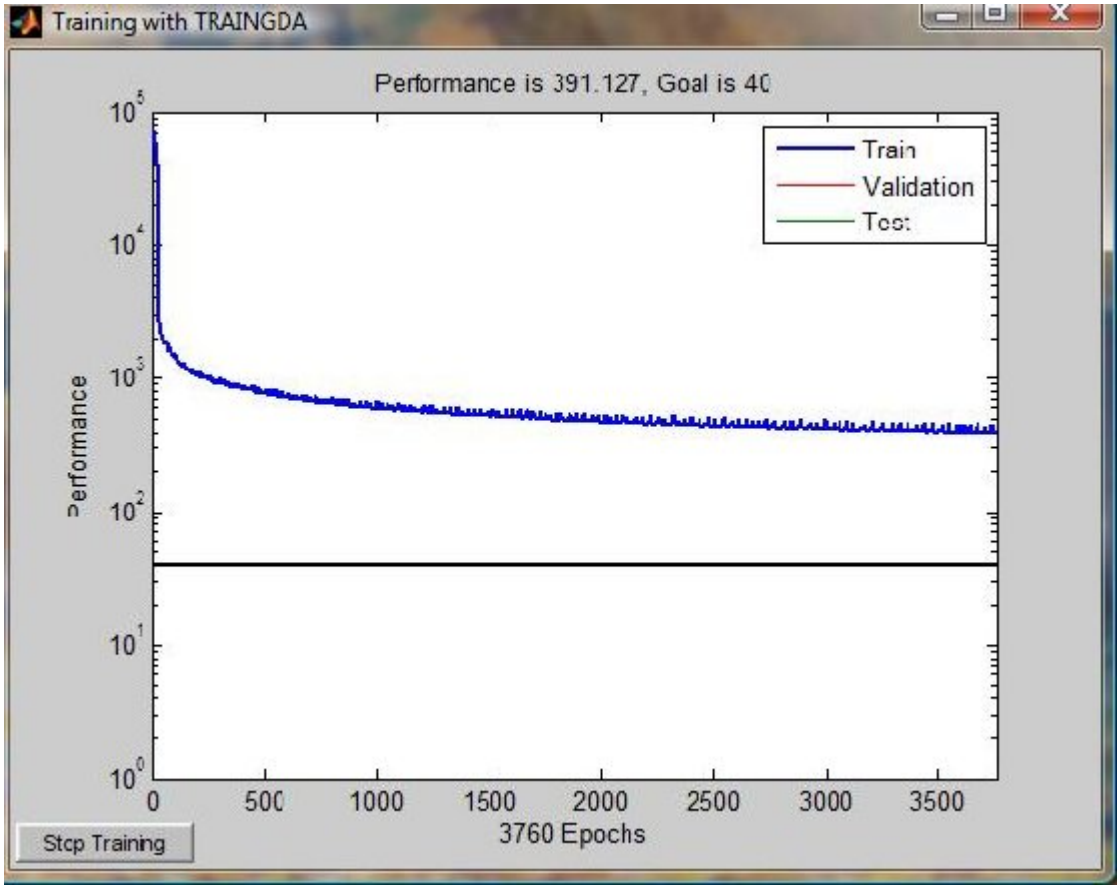


Figure 6.59 Output of training dataset using Traingda Training Function

The graph shown in figure 6.59 represents the output of the training of the network and 3760 epochs have been taken to get trained the network using the traingda train function. In this case the performance goal of the network has been 391.127.

Figure 6.60 Original and Decompressed Image of cameraman after 3760 epochs

Table 6.3 Different values of CR, RMSE, PSNR, B/P taken at different epochs

EPOCH	CR	RMSE	PSNR	B/P
1100	.9912	27.1196	19.3181	.5881
1202	.9864	26.8815	19.6821	.5952
1300	.9802	26.7192	19.9852	.6103
1340	.9755	26.4473	20.1531	.6209
1620	.9689	26.1217	20.3594	.6351
1900	.9612	25.9891	20.5893	.6493
2200	.9567	25.8371	20.7231	.6537
2950	.9517	25.6192	20.9817	.6681
3380	.9455	25.4156	21.1432	.6754
3760	.9387	25.2189	21.4102	.6789

The above values of CR, RMSE, PSNR shows that image is compressed with very low loss of image quality. As the values of epochs is increasing from 1100 epochs to 3760 epochs compression ratio have been decreased from .9912 to .9387 and Peak signal to noise ratio has been increased from 19.3181 to 21.4102 . This is because of network is getting more time to adjust their weight and more optimized weight are obtained to train the network.

Chapter 7

Conclusion and Future Scope

Conclusion

The implementation of back propagation neural network algorithm on image compression system with good performance has been demonstrated. The back propagation neural network has been trained and tested for the analysis of different images. It has been observed that the convergence time for the training of back propagation neural network is very faster. Different attributes of compression such as compression ratio, peak signal to noise ratio, bits per pixel are calculated. It has been observed that there is significance change in compression ratio from .9972 to .9461 in case of Lena and from .9965 to .9417 in case of Girl and from .9912 to .9387 in case of Cameraman image. It has also been observed that there is significance improvement in peak signal to noise ratio from 20.5966 to 22.1725 in case of Lena and from 19.2283 to 20.5696 in case of Girl and from 19.3181 to 21.4102 in case Cameraman. The adaptive characteristics of the proposed approach provide modularity in structuring the architecture of the network, which not only speed up the processing but also less susceptible to failure and easy for rectification. Instead of generating multiple training patterns and imparting off-line training, here due to the considerable reduction of image size only single training pattern is used to train the NN and on-line training could be invoked with practical implication of the system. The technique of initialization of weights exhibits fast rate of convergence and using the trained weight sets, good quality of regenerated images are available at the receiving end.

Future Scope

The field of image processing has been growing at a very fast pace. The day to day emerging technology requires more and more revolution and evolution in the image processing field. As showed in this thesis work, back propagation neural networks can be successfully to implement the image processing. The same experiments should also be conducted with other types of neural network to see the different types can improve the performance of the system as we got the experiments results with the back propagation neural network.

References

[1] Abbas Razavi, Rutie Adar, Isaac Shenberg, Rafi Retter and Rami Friedlander "VLSI implementation of an image compression algorithm with a new bit rate control capability" Zoran Corporation, 1705 Wyatt Drive, Santa Clara, CA 95054.This paper was published in IEEE international conference on 26th march 1992 on volume 5 and pages 669-72.

[2] Ronald A.DeVore, Bjorn Jawerth, and Bradley J.Lucier. "Image compression through wavelet transforms coding" IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 38. NO .2, MARCH 1992.

[3] David Jeff Jackson and Sidney Jwl Hannah "Comparative analysis of image compression technique" Department of Electrical Engineering ,The University of Alabama, Tuscaloosa, AL 35487.This paper appears in system theory 1993,proceeding SSSST'93, twenty fifth edition southeastern symposium on 9th march 1993 on pages 513-517.

[4] P.Moravie, H.Essafi, C. Lambertt-Nebout and J-L. Basill. "Real time image compression using SIMD architecture" Centre Spatial de Toulouse 18 Avenue Edouard Belin BP 1421. This paper appears in computer architecture for machine perception on 20 September 1995 on pages 274-279.

[5] J Jiang "Neural network technology for image compression" Bolton Institute, UK. This paper appears in broadcasting convection 1995,IBC 95,on 18 September 1995 on pages 250-257.

[6] Wayne O. Cochran, John C. Hart "Fractional volume compression". Member of IEEE computer society .This paper appears in visualization and computer graphics IEEE, transactions on December 1996 on pages 313-322.

[7] Maire D.Reavy and Charles G.Boncelet "BACIC: a new method for lossless bi-level and gray scale image compression". Electrical Engineering Department, University of Delaware Newark, DE I9716. This paper appears in Image processing, 1997, proceedings, International conferences on 29th October 1997 on volume 2 pages 282- 285.

[8] S.W.Hong and P.Bao "Hybrid image compression model based on sub band coding and edge preserving regularization. This paper was published on IEEE proc-vis. Image signal process volume 147, February 2000.

[9] Mitchell A. Golner, Wasfy B. Mikhael and Venkatesh Krishnan."Region based variable quantization for JPEG image compression". School of ElectricalEngineering and Computer Science,University of Central Florida Orlando, FL 32816-2450. This paper appears in circuit and system 2000, proceeding of the 43rd IEEE Midwest symposium on volume 2 , pages 604 -607.

[10] Clark N. Taylor and Sujit Dey "Adaptive image compression for wireless multimedia communication" Electrical and Computer Engineering University of California, San Diego La Jolla, California, USA .This paper appears in communication, 2001 .ICC 2001,IEEE International conference on 14th June 2001, volume 6 on pages 1925 -1926.

[11] Pou-Yah Wu ."Distributed fractal image compression on PVM for million pixel images". Department of Mathematics Education National Tainan Teachers College, Tainan 700 Taiwan. This paper was presented on 15th International conference on information networking in 2001 on page 393.

[12] Koon-Pong Wong "Fractal image coding for emission topographic image compression". Department of PET and Nuclear Medicine Royal Prince Alfred Hospital, NSW 2050, Sydney, Australia. This paper was published in Nuclear science symposium conference 2002 IEEE .

[13] Michael T.Kurdziel. "Image compression and transmission for HF radio system". Harrish corporation RF communication division Rochester, NY. This paper appears in MILLCON 2002, volume 2 on pages 1281-1285.

[14] Aaron T. Deever and Sheila S."Lossless image compression with projection based and adaptive reversible interger wavelet transform". This paper appears in IEEE transactions of image processing, volume 12, on May 2003.

[15] Deepti Gupta, Shital Mutha. "Image compression using wavelet packet". Students-Cummins College of Engineering, Karvenagar, Pune-52. This paper was published in IEEE conference in 2003.

[16] G. Y. Chen, T. D. Bui and A. Krzyzak "Image compression with optimal wavelet". Department of Computer Science Concordia University 1455 De Maisonneuve West Montreal,

Quebec, Canada H3G IM8 .This paper appears in electrical and computer engineering 2004 , Canadian conference on 5th May 2004 , volume 1, pages 209-212.

[17] Mei Tian, Si-Wel Luo and Ling-Zhi Liao " An investigation into using singular value decomposition as a method of image compression ".School of Computer and Information Technology , Beijing Jiaotong University, Beijing 100044, China . This paper appears in Machine learning and cybernetics 2005, Proceeding of 2005 International conference on 21st August 2005 , volume 8 on pages 5200-5204.

[18] Erjun Zhao and Dan Liu. "Fractal image compression method review". Institute of Nautical Science & Technology Dalian Maritime University, Dalian 116026, P.R.China. This paper appears in information technology and application, 2005. ICITTA 2005, Third International conference on 7th July 2005 , volume 1 on pages 756 – 759.

[19] Sheng Liu, Na Jiang, Like Zhang and Donghao Xu ."Research on the radar image compression of VDR based oh SPIHT". College of Automation Harbin Engineering University Harbin Heilongjiang Province, China. This paper appears in Mechatronics and automation ,proceeding of 2006 IEEE International conference on 28th June 2006 on pages 2357-2361.

[20] Xiangjian He "Fractal image compression on spiral architecture." Computer Vision Research Group, University of Technology, Sydney. This paper appears computer graphics, imaging and visualization, 2006 international conference on 26th July 2006 on pages 76-83.

[21] Ivan Vilovic. "An experience in image compression using neural network". University of Dubrovnik. This paper appears in multimedia signal processing and communication, 48th International symposium ELMAR -2006 focused on June 2006 on pages 95-98.

[22] Tao Wang, Dongmei Li, Chunkuang Tao, Haiquan Shi "Research of image compression based on optical wavelet transform". College of Mathematics and Physics, Chongqing University. Chongqing, 400044, China .This paper appears in control, automation, robotics and vision 2006. ICARCV 06. 9th International conference on 8th December 2006 on pages 1-5.

[23] Kin-Wah Ching Eugene and Ghim-Hwee Ong. " A two pass improved encoding scheme for fractional image compression". Kin-Wah Ching Eugene and Ghim-Hwee Ong, School of Computing, National University of Singapore. This paper appears in computer graphics, image and visualization, 2006 International conference on 28th July 2006 on pages 214-219.

[24] Yukinari Nishikawa, Shoji Kawahito, Masanori Furuta and Toshihiro Tamura " A high speed CMOS image sensor with ON-chip parallel image compression circuit ". Graduate School of Electronics Science and Technology Shizuoka University. This paper appears in custom integrated circuit conference on 19th September 2007.

[25] Jian Li, Caixin Sun. "Partial discharge image recognition influced by fractal image compression**".** Department of High Voltage and Insulation Technology, College of Electrical Engineering, Chongqing University. China. This paper appears in Dielectric and electrical insulation, IEEE transactions on April 2008, volume 15, pages 496-504